

## Homework #3

Due Time: 2018/12/13 (Thu.) 14:20

Contact TAs: [ada-ta@csie.ntu.edu.tw](mailto:ada-ta@csie.ntu.edu.tw)

### Instructions and Announcements

- There are **four programming problems** and **two hand-written problems**, and the homework set including bonus are worthy of 103 points. If you get more than 100 points, your score will still be counted as 100 points.
- **Programming.** The judge system is located at <https://ada18-judge.csie.org>. Please login and submit your code for the programming problems (i.e., those containing “Programming” in the problem title) by the deadline. **NO LATE SUBMISSION IS ALLOWED.**
- **Hand-written.** For other problems (also known as the “hand-written problems”), you **MUST** turn in a **printed/written version** of your answers to the instructor at the beginning of the class on 2018/12/13. **Remember to print your name/student ID on the first page of your submitted answers.** You can also upload your homework to the NTU COOL system; however, it will be marked **only when** you have turned in the printed/written answer but it is lost during the grading. **NO LATE SUBMISSION IS ALLOWED.**
- **Collaboration policy.** Discussions with others are strongly encouraged. However, you should write down your solutions **in your own words**. In addition, for **each and every** problem you have to specify the references (e.g., the Internet URL you consulted with or the people you discussed with) on the first page of your solution to that problem. You may get zero point due to the lack of references.
- With TAs’ discretion, too complicated solutions will be counted as wrong.

## Problem 1 - Teddy Bears' Friendships (Programming) (10 points)

### Problem Description

In homework 2 problem 4, the problem description said that “You have heard of ADA Kingdom in homework 1”. This is not true, for that the ADA Kingdom is so mysterious that it didn't appear in homework 1. Today I'm going to tell you a story about ADA kingdom.

After eliminating the anti-handsome stuffs in homework 2, Handsome decide to expand his territory. The ADA Kingdom is located at the origin of a 1-dimension world and Handsome had conquered half of the world (i.e.,  $[0, \infty)$ , for that it is a 1D world.) with his power of vector field.

Handsome has  $n$  teddy bears that can control his kingdom, and he wants to maximize the range of his territory by assigning his teddy bears as far as possible. The range of his territory is defined by the most furthest teddy bear's coordinate.

However, the teddy bears can not be too far from their friends. We can use  $m$  “friendship tuples” to describe their friendships: a 3-tuple  $(a, b, c)$  means that the distance between teddy bear  $a$  and teddy bear  $b$  should never exceed  $c$ . That is, if the two teddy bears are located at  $x_a$  and  $x_b$ , respectively, then  $|x_a - x_b| \leq c$  must hold. In addition, the teddy bear 1 (Handsome's favorite teddy bear: Linear Algebra King) must stay at the capital (the origin point) of the ADA kingdom.

Can you help Handsome find out that what is the maximum territory it can reach without destroy the teddy bears' friendships?

### Input

The first line contains 2 integers  $n, m$  indicating the number of the teddy bears and the “friendship tuples”, respectively.

For the next  $m$  lines, each of them contains 3 integers  $a, b, c$  indicating the constraints between the specified teddy bears.

- $1 \leq n \leq 100,000$
- $0 \leq m \leq 300,000$
- $1 \leq a, b \leq n$
- $0 \leq c \leq 1,000,000$
- Each pair  $a, b$  appears at most once and  $a \neq b$ .

### Output

Please output a integer indicating the maximum value of ADA Kingdom's territory. If the value can be arbitrarily large, please output “-1” (without quotes).

#### Sample Input

```
4 4
1 2 2
1 3 4
2 3 1
3 4 5
```

#### Sample Output

```
8
```

**Hint**

In shortest path algorithm, let  $d_u$  is the shortest distance from a given source to  $u$  after solving a single source shortest path problem. Then if there is a weighted undirected edge that connects  $(u, v)$  with weight  $c$ ,  $d_u + c \geq d_v$  and  $d_v + c \geq d_u$  holds.

The inequality can be rewritten as  $|d_v - d_u| \leq c$ . That is, if we construct a weighted undirected graph with the “friendship tuples”, the distance array obtained from shortest path algorithms is a solution of the inequalities. Because all the initial distances are set to  $\infty$ , the shortest path solution should be the maximum value subject to the constraints.

## Problem 2 - The Great Wall (Programming) (10 points)

### Problem Description

Joe is one of the most brutal emperor in ancient China. One day, Kevin, Joe's military adviser, tells him a plan in order to enlarge the territory of his empire. He decides to build a great wall to avoid the reckless attacks and casualties. The great wall consists of several segments, with each segment having a troop to defend the enemy.

Let's say that the great wall is on a one dimensional line and it is composed of sets of disjoint segments. Each segment  $[l_i, r_i]$  has a power  $(r_i - l_i)^2$ . The power of the great wall is the summation of all segments' power.

The construction process takes  $n$  days. Every morning, Kevin assigns a new segment  $[l_i, r_i]$  to build. All segments intersecting with the new one will be combined to a single segment, even if they only have one common point on the line. In the evening, Kevin will record the new power of the great wall. However, calculating the power is not an easy thing, so Kevin asks you to write a program that automatically calculates the total power of the great wall.

Note that  $r_i$  is greater than  $r_{i-1}$  for every  $i$ .

### Input

The first line of input contains an integer  $n$  indicating the number of days the process took.

Each of the following  $n$  lines has two integer  $l_i, r_i$  indicating the assigned segment on the  $i^{th}$  day.

- $1 \leq n \leq 200,000$
- $1 \leq l_i \leq r_i \leq 10^9$

### Output

For each day, output an integer indicating the power of great wall in the evening.

### Subtask 1 (40 %)

- $n \leq 1000$

### Subtask 2 (60 %)

- no other constraints.

### Sample Input 1

```
3
1 3
4 6
5 7
```

### Sample Output 1

```
4
8
13
```

## Problem 3 - String World (Programming) (15 points)

### Problem Description

Imagine a world equipped with character set  $\mathbf{U} = \{c_1, c_2, \dots, c_k\}$  with unknown total order  $\prec_{\mathbf{U}}$ . You are given an important string  $S = \langle s_1 s_2 \dots s_n \rangle$  consists of characters from set  $\mathbf{U}$ . Also, the lexicographical order of **suffices** of  $S$  is given. Please write a program to compute a possible total order  $\prec'_{\mathbf{U}}$  (maybe not identical to  $\prec_{\mathbf{U}}$ ) of  $\mathbf{U}$ .

$S[i, j]$  with  $1 \leq i \leq j \leq |S|$  is the string  $\langle s_i s_{i+1} \dots s_j \rangle$  where  $|S|$  is length of  $S$ . **Suffices** of  $S$  is the set of strings  $\mathbf{suf}(S) = \{S[i, |S|] \mid 1 \leq i \leq |S|\}$ . For example, suffices of string "abb" are "abb", "bb" and "b".

String  $X$  is lexicographically smaller than string  $Y$  (written as  $X \prec_{\mathbf{U}}^* Y$ ) if either  $X$  is exactly  $Y[1, |X|]$  or  $X_i \prec_{\mathbf{U}} Y_i$  with  $i$  being the first position with different character. For example, consider the character set of lowercase letters with common character order, "aa" is lexicographically smaller than "aab" and "acc" is lexicographically smaller than "baa".

In this problem you are given the rank of each suffix of  $S$ . The smaller the rank is, the smaller the lexicographical order is. Suppose we are given the order that b is smaller than a. String "abba" has its smallest suffix being "bba" and the biggest one being "abba". To construct the rank one may have multiple choices for order of  $\mathbf{U}$ . Both  $\mathbf{b} < \mathbf{c} < \mathbf{a}$  or  $\mathbf{c} < \mathbf{b} < \mathbf{a}$  act as a valid choice.

Suffix	Rank
"abba"	4
"bba"	1
"ba"	2
"a"	3

Table 1: Rank of suffices of "abba"

### Input

The first line of input contains an integer  $T$  indicating the number of test cases.

For each test case, there are 3 lines of input. First line of the test case has two space-separated integers  $n, k$ . Second line of the test case has  $n$  space-separated integers  $a_1, a_2, \dots, a_n$  representing the important string  $S = \langle c_{a_1} c_{a_2} \dots c_{a_n} \rangle$ . Third line of the test case has  $n$  space-separated distinct integers  $r_1, r_2, \dots, r_n$  representing the rank of each suffix of  $S$ .  $r_i$  is the rank of  $S[i, n]$  among  $\mathbf{suf}(S)$ .

- $1 \leq T \leq 100$
- $2 \leq n, k \leq 200000$
- $1 \leq a_i \leq k$  for every  $i = 1, 2, \dots, n$
- $1 \leq r_i \leq n$

### Output

For each test case, please output one line of  $k$  space-separated integers  $b_1, b_2, \dots, b_k$  with  $c_{b_1}$  being the smallest character of  $\mathbf{U}$ ,  $c_{b_2}$  the second smallest and so on.

There may be multiple possible solutions. Your solution is considered correct if ranks of suffices of  $S$  don't change using your character order. There must be a valid solution for every test case.

**Subtask 1 (30 %)**

- $k = 2$

**Subtask 2 (40 %)**

- $n, k \leq 5000$

**Subtask 3 (30 %)**

- no other constraints.

**Output**

For each test case, please output one line of  $k$  space-separated integers  $b_1, b_2, \dots, b_k$  with  $c_{b_1}$  being the smallest character of  $\mathbf{U}$ ,  $c_{b_2}$  the second smallest and so on.

There may be multiple possible solutions. Your solution is considered correct if ranks of suffices of  $S$  don't change using the character order you output.

By the way, there must be a valid solution for every test case.

**Sample Input 1**

```
2
4 2
1 2 2 1
4 1 2 3
3 3
1 1 1
3 2 1
```

**Sample Input 2**

```
2
3 3
1 2 3
1 2 3
3 3
2 1 3
1 2 3
```

**Sample Output 1**

```
2 1
3 2 1
```

**Sample Output 2**

```
1 2 3
2 1 3
```

## Problem 4 - Casino (Programming) (15 points)

### Problem Description

You went into a casino. There's a huge, mysterious machine. To get the prize inside the machine, you are required to input an  $N$ -bit password  $p_1p_2p_3\dots p_N$ , and  $p_i \in \{0, 1\}$  for every  $i$ . However, if the password you input is wrong, then the machine will self-destruct immediately, and you can no longer get any prize.

You are allowed to ask some questions to the machine. In each question, you can pick an interval  $[l, r]$ , and the machine will give you the value  $\bigoplus_{i=l}^r p_i$ . It takes  $t_{l,r}$  for the machine to answer such question. Initially, there's no prize inside the machine, and each such question will increase the prize by  $a_{l,r}$ . Your goal is to ask as few questions as possible, and to maximize  $\frac{\Sigma \text{prize}}{\Sigma \text{time cost}}$  in that situation. Now, you are give all those  $t_{l,r}$  and  $a_{l,r}$ , what is the maximum  $\frac{\Sigma \text{prize}}{\Sigma \text{time cost}}$  you can get when asking the least questions?

$\bigoplus$  means XOR operator.

### Input

The first line of input contains an integer  $N$  indicating bits of passwords.

The following  $N$  lines, the  $i + 1$  line has  $N - i - 1$  integer  $a_{i,j}$  indicating the the prize when asking  $[i, j]$ .

The following  $N$  lines, the  $i + 1$  line has  $N - i - 1$  integer  $t_{i,j}$  indicating the the time cost when asking  $[i, j]$ .

- $1 \leq N \leq 1024$
- $1 \leq a_{i,j}, t_{i,j} \leq 10^5$

### Output

Output two integer  $P$  and  $Q$  after fraction reduction, which  $\frac{P}{Q} = \frac{\Sigma \text{prize}}{\Sigma \text{time cost}}$

#### Subtask 1 (30 %)

- $N \leq 128$
- $a_{i,j} = 1$

#### Subtask 2 (70 %)

- $a_{i,j} = 1$

#### Subtask 3 (bonus) (20 %)

- no other constraints.

**Sample Input 1**

```
5
1 1 1 1 1
1 1 1 1
1 1 1
1 1
1
1 2 3 4 5
4 3 2 1
3 4 5
2 1
5
```

**Sample Output 1**

```
5 7
```

**Sample Input 2**

```
5
2 1 1 1 1
1 1 1 2
1 1 1
1 1
1
1 2 3 4 5
4 3 2 1
3 4 5
2 1
5
```

**Sample Output 2**

```
1 1
```



## Problem 5 - Saiko Psycho-Pie (Hand-Written) (25 points)

### Story

Once upon a time, in the ADA kingdom, there was a young brave knight named Piepie. At the other side of the kingdom, lived an ambitious sorcerer called QMao.

QMao the sorcerer was just one ordinary magician until one day, he came upon the brilliant idea of creating a special treat – psycho-pie.

Psycho-pies were no ordinary pies that can be found in bakeries. In fact, rumours had that it's flavour was so capturing, that once tasted, one can no longer bear to live without it. One single bite of the Psycho-pie could make any average human lose appetite for any other food. If not provided with more Psycho-pies, the crave for another bite would result in some kind of withdrawal symptoms and eventually drive one mad.

As mentioned earlier, QMao was ambitious, and would do anything to gain power. Seeing his chance, he soon started manufacturing and spreading Psycho-pies all over the continent. It wasn't long before Psycho-pies became the most prevalent drug in the ADA kingdom.

The king of ADA kingdom was a kind old man – Handsome (details in problem 1), and could not allow his subjects to suffer from the harm of Psycho-pies. He demanded the Mage Academy to investigate on the Psycho-pies, and soon reached the conclusion that Psycho-pies are just ordinary pies with mind-control magic casted on them. No one knows how a self-taught second-class mage like QMao had obtained such power, but it was clear not even the best mages in the Academy can weave a spell strong enough to counter it.

The last hope of stopping QMao is by assassinating him, for this single cause, Piepie the Valiant was brought before the old king.

A single order was given, bring justice back to the continent.

As Piepie prepared to depart, another bad news shook the entire kingdom. Multiple cities had been captured by QMao's followers. Travellers who come near such cities were forced to eat Psycho-pies and declare loyalty to QMao.

Things have gotten out of control.

True enough, Piepie is by far the strongest warrior in ADA kingdom(probably the strongest throughout the entire history), but not even he can fend off attacks from brain-washed mobs. Nor can he withstand the curses casted on Psycho-pies. Travelling across the kingdom was no longer an easy task.

The Imperial Army and Mage Academy both offered to provide help, but more important than this, Piepie needed a plan. A plan that can bring him across the now chaotic continent and hunt down QMao.

You, one of Piepie's best friend, must help come up with such a plan before everything is too late. The future of ADA kingdom, justice, and the world is in your hands.

### Problem Description

- The map of ADA kingdom is a directed graph, with cities as vertices and roads as weighted edges.  $V$  denotes the set of cities, while  $E$  denotes the set of weighted edges.
- Position of the empire city(where Piepie starts his journey) , and QMao's workhouse are fixed.
- Piepie is strong enough to endure the curse of Psycho-pies for a small interval of time. If unfortunately forced to eat any Psycho-pies, he is capable of travelling another  $k$  cities before

needing to consume another piece of Psycho-pie to suppress withdrawal symptoms.

- Unlimited amount of Psycho-pies are provided in cities captures by QMao. However, it is prohibited to carry any Psycho-pies as travellers leave those cities.
- The scenarios proposed in subproblem (d),(e),(f),(g) are independent from each other, and should be considered separately.
- For simplicity,  $dijkstra(V, E)$  can be used to denote the time complexity for finding the shortest path from source to target in a graph with  $V$  and  $E$ .
- For the problem below, please simply prove the *correctness* for each subproblem. Grades will also be awarded to non-correct answers basing on the algorithm complexity and prove of correctness of your algorithm.

### Grading Criteria

- Subproblem (a),(b) is optional. If you are confident enough with your answer of subproblem (c), you can choose to not write subproblem (a),(b). The points for subproblem (a),(b) will then be transferred to subproblem (c).
- For subproblem (d),(e),(f),(g) ,you have to choose one to write from the four combinations (d,f),(d,g),(e,f),(e,g), and yes, you need only write two subproblems. If more than two subproblems were written, points will be calculated based on the combination with highest grade, but no additional bonus will be awarded.

#### 5-(a) (3 points) - No-pie-no-pie-no-piepie Strike 1

In this subproblem,  $k = 2$ .

Before departure, some traitor mixed Psycho-pies into Piepie's dinner and successfully drugged him. Please provide an algorithm that finds the shortest route for Piepie to reach QMao's home alive. Your algorithm should run in  $O(dijkstra(V, E))$ .

#### 5-(b) (4 points) - No-pie-no-pie-no-piepie Strike 2

In this subproblem,  $k = 2$ .

Piepie is careful enough to not be drugged by traitors before departure. Please provide an algorithm that finds the shortest route for Piepie to reach QMao's home alive. Your algorithm should run in  $O(dijkstra(V, E))$ .

#### 5-(c) (6 points)

Consider subproblem (2) under the general case where  $k$  might not be 2. Your algorithm should run in  $O(k \times dijkstra(V, E))$ .

Hint: Duplicate all vertices  $k$  times, each representing how much time is left before Piepie must consume another piece of Psycho-pie.

#### 5-(d) (6 points) - TheraPIE

As the old saying goes, fight poison with poison. Archimage Masschian has developed a spell to duplicate cursed Psycho-pies. He named the resulting product TheraPIE. By consuming TheraPIEs,

Piepie can soothe his craving for Psycho-pie and stand another  $k$  time before needing to consume another piece of Psycho-pie. However, producing TheraPIEs is not easy, it takes tens of months of continuous enchanting for even the most skillful mage to weave the spell to produce a single piece of TheraPIE. Before Piepie's departure, the mages were only able to craft a single piece of TheraPIE.

Now, having received the gift from those mages, please provide an algorithm that utilizes TheraPIE and finds the shortest route for Piepie to reach QMao's home alive. Your algorithm should run in  $O(k \times \text{dijkstra}(V, E))$ .

### 5-(e) (6 points) - Hyperdimensional Shift

Relics left from the ancient DSA kingdom have the power to teleport people to specific places. Each relic has one specific teleportation target. Piepie the Valiant was informed by archimage Masschian of the positions and the teleportation target of each relic. However, the relics are old, and need to be activated before being used. Since activating the relics demands great skills and is an extremely wearisome task, Masschian can only help activate one relic for Piepie.

Given this additional information, please provide an algorithm that utilizes those relics and finds the shortest route for Piepie to reach QMao's home alive. Your algorithm should run in  $O(k \times \text{dijkstra}(V, E) + E_2)$ .

### 5-(f) (6 points) - Imperial March

While Piepie continues on his journey, the Imperial Army also tries their best to support him. General Nivra has promised Piepie to recover certain cities. However, troops are expected to suffer serious damage in encountering the brain washed mobs, thus, only a maximum of  $k_2$  cities can be recovered. Please provide an algorithm that utilizes the recovery of those  $k_2$  cities and finds the shortest route for Piepie to reach QMao's home alive. Your algorithm should run in  $O((k+k_2) \times \text{dijkstra}(V, E))$ .

### 5-(g) (6 points) - Nirvana

One old technique passed down through General Nivra's family is Nirvana. Practicers of Nirvana has the ability to temporarily boost physical abilities and freeze body conditions. However, Nirvana can be used only once in a life.

Having learned this technique, upon activated, Piepie will break through the next continuous  $k_2$  cities without being forced to eat Psycho-pies (However, note that during activation of Nirvana, Piepie cannot consume Psycho-pies even if he wanted to). During this time, the countdown before needing to consume another piece of Psycho-pie will also be frozen.

Now, please provide an algorithm that utilizes the technique Nirvana and finds the shortest route for Piepie to reach QMao's home alive. Your algorithm should run in  $O(k \times k_2 \times \text{dijkstra}(V, E))$ .

### Problem 6 - The Infinity Ward (Hand-Written) (25 points)

Riot Games, the developer of League of Legends (LoL), is going to publicize a new unit called Infinity Ward. After deployed, the Infinity Ward allows the player to have an infinite range of sight around the unit unless blocked by any obstacles. Eddy, an iron tier player, is excited about this update because he believes he can come up with a perfect idea to maximize the benefit of Infinity Ward, and eventually reaches diamond tier. The original map of LoL is a simple polygon with holes, but we consider a simplified version: a simple polygon without any hole.

Eddy formulates the problem as following:

*A set  $S$  consists of Infinity Wards at different positions in a simple polygon  $P$  with  $n$  edges. Find the upper bound of the size of  $S$  given  $n$ , such that every point  $p \in P$  can be visible by at least one Infinity Ward  $s \in S$ . Two points are visible from each other if the line segment between them does not leave the polygon.*

For example, in Figure 1a, only one Infinity Ward is needed for every point in the polygon. However, one is not enough for some polygons, such as the one shown in Figure 1b. Apparently, the polygon in Figure 1b needs two Infinity Wards to guard.

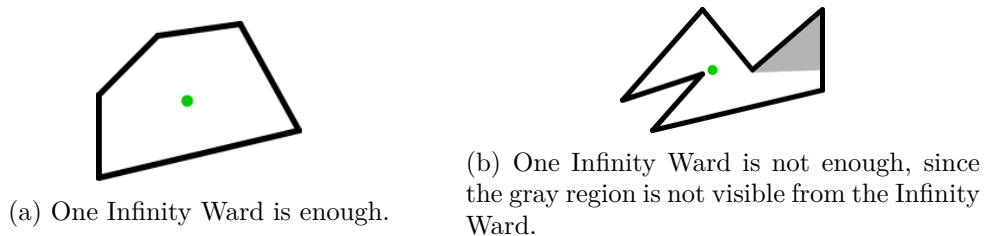


Figure 1: Black lines form the polygon  $P$ , green dots are the set  $S$  of Infinity Wards.

In order to solve this problem, let's first introduce a term: outerplanar graph.

*An outerplanar graph is a graph that has a planar drawing for which all vertices are not surrounded by other vertices.*

For example, Figure 2a is an outerplanar graph, whereas Figure 2b, which is  $K_4$ , is not an outerplanar graph no matter how you draw it.

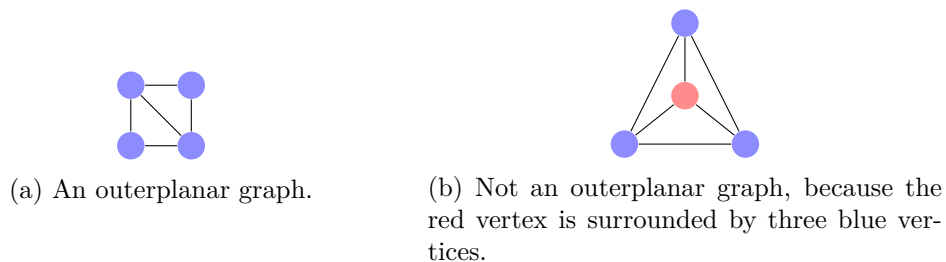


Figure 2: Examples of an outerplanar graph and a non-outerplanar graph.

Some outerplanar graphs remain outerplanar even if you add some edges to it. For instance, you can add edge (0,3) and edge (1,4) to the outerplanar graph in Figure 3a, and the new graph is still outerplanar. However, you cannot add more edge to Figure 3b, or else it will become non-outerplanar.

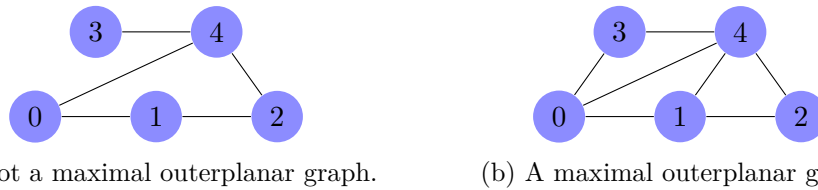


Figure 3: Examples of a non-maximal outerplanar graph and a maximal outerplanar graph.

Now, you are able to solve the following questions sequentially, and help Eddy reaches diamond tier!

**(a) (5 points)**

Please prove the following statement:

*Outerplanar graph with  $n$  vertices has at most  $2n - 3$  edges.*

Hint:  $V - E + F = 2$  holds for any planar graph, where  $V$  is the number of vertices,  $E$  is the number of edges, and  $F$  is the number of faces (regions), including the exterior face. As an illustration,  $K_2$  has one face,  $K_3$  has two faces, and  $K_4$  has four faces.

**(b) (5 points)**

Please prove the following statement:

*Outerplanar graph is 3-colorable. That is, 3 colors are sufficient to color all vertices of the graph such that no neighboring vertices have the same color.*

Hint: You can directly employ the statement in subproblem (a) in your proof.

**(c) (4 points)**

Please write down the pseudo-code of a fastest algorithm that can color outerplanar graphs in 3 colors.

**(d) (2 points)**

Please prove the time complexity of your algorithm in subproblem (c).

**(e) (8 points)**

Derive the answer to the original problem and prove it.

Hint: The answer is  $\lfloor \frac{n}{3} \rfloor$ .

**(f) (1 point)**

Draw an actual polygon to show that your answer in subproblem (e) is the tightest *upper bound*.