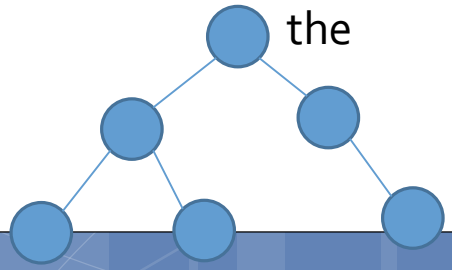
Dynamic Programming III

Michael Tsai

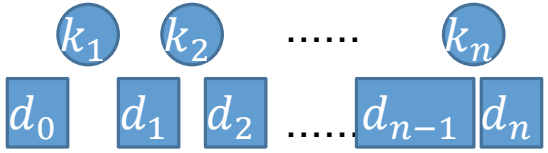
2013/10/11

翻譯機問題

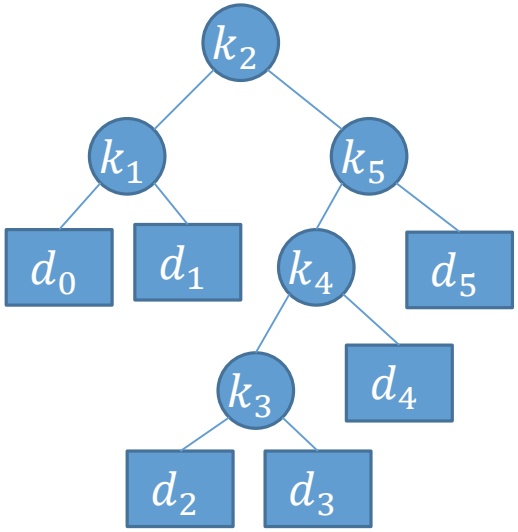
- 最笨翻譯機:
  每個英文單字直接翻成法文單字
- 做法: 建一棵balanced binary search tree (例如紅黑樹), 裡面用英文單字當key, 法文單字當作對應的資料
- 則每個字平均花$O(\log n)$的時間
- 假設我們知道每個字出現的頻率(或機率), 可以做得更好嗎?
- 答: 可以! 把常用的字放離root近一點.



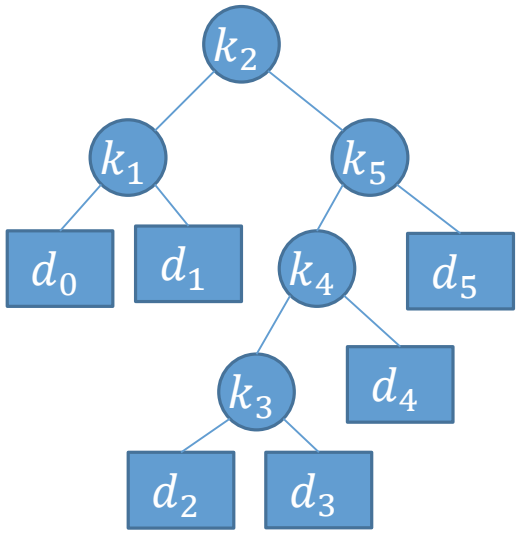
翻譯機問題→Optimal Binary Search Tree



- 問題:
  - 給一個序列$K = \langle k_1, k_2, \ldots, k_n \rangle$共n個排好序的key ($k_1 < k_2 < \cdots < k_n$). 我們要用這些key建立一棵binary search tree.
  - k_i出現的機率為p_i.
  - 另外我們也有n+1個"假key"代表沒有出現在K中的值, 可用$d_0, d_1, d_2, \ldots, d_n$來表示. d_0代表小於k_1的值, d_1代表介於k_1和k_2的值,...,d_n代表大於k_n的值.
  - 假key d_i出現的機率為q_i.
  - 則目標是找出一棵binary search tree使得Expected cost最小.



- $E[search\ cost\ ] = \sum_{i=1}^{n}(depth_T(k_i) + 1) \cdot p_i + \sum_{i=0}^{n}(depth_T(d_i) + 1) \cdot q_i$
  $= 1 + \sum_{i=1}^{n} depth_T(k_i) \cdot p_i + \sum_{i=0}^{n} depth_T(d_i) \cdot q_i$
- since $\sum_{i=1}^{n} p_i + \sum_{i=0}^{n} q_i = 1$
- 使得以上E[search cost]最小的binary search tree稱為optimal binary search tree.



i	0	1	2	3	4	5
p_i		0.15	0.10	0.05	0.10	0.20
q_i	0.05	0.10	0.05	0.05	0.05	0.10

- 假設給定的key的出現機率為右上表格所顯示, 則左上圖為optimal binary search tree (expected cost=2.75)
- 觀察: 機率最大的key不見得在root (k_5不在root)

暴力法有多暴力?

- n個node的binary search tree總共有$\Omega(\frac{4^n}{n^{\frac{3}{2}}})$個(Catalan number)

Dynamic Programming出招

1. 找出Optimal Substructure

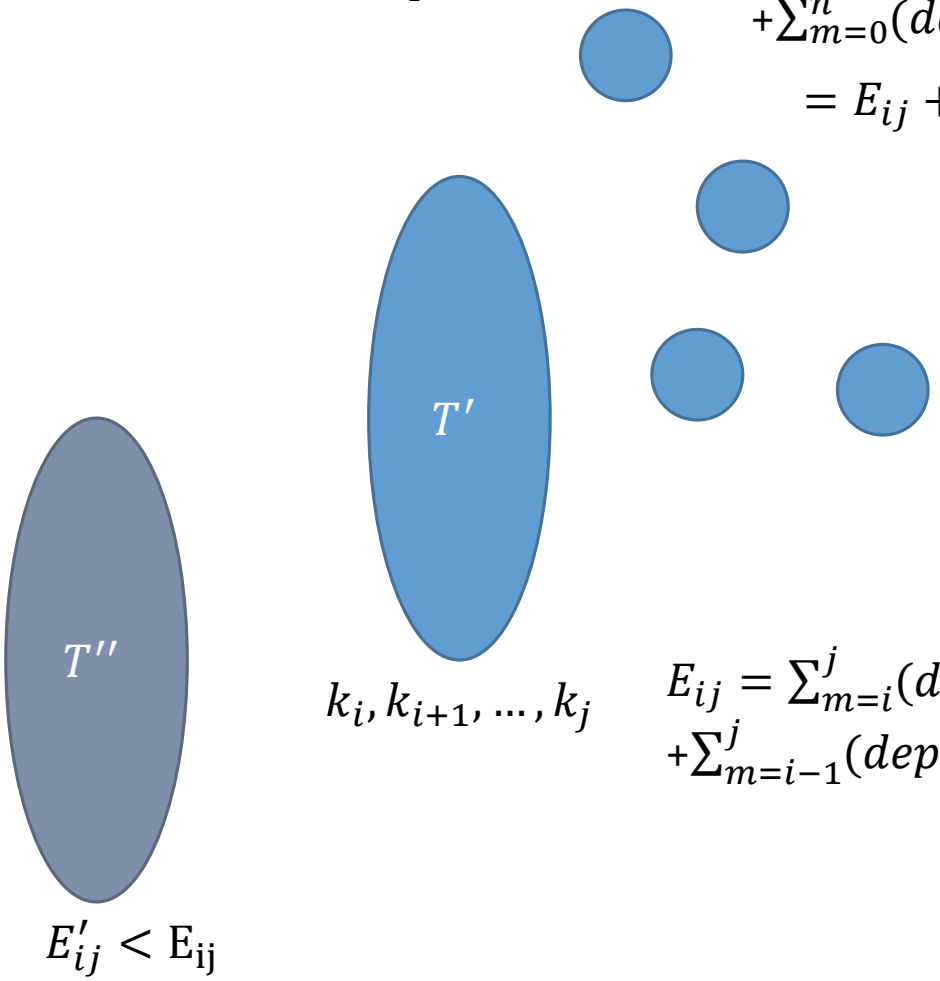
- 小觀察: binary search tree的subtree必包含一段連續的key $k_i, k_{i+1}, \dots, k_j$及$d_{i-1}, \dots, d_j$, $1 \le i \le j \le n$.
- 小定理: 假設T為$K = \langle k_1, k_2, \dots, k_n \rangle$之optimal binary search tree. 則T之subtree T',包含$k_i, \dots, k_j$這些key, 也必定是$K' = \langle k_i, k_{i+1}, \dots, k_j \rangle$這些key的optimal binary search tree.
- 證明: 如果T'找出的不是optimal binary search tree, 則表示可以找出一個更好的binary search tree T'', expected cost比T'更好, 則可以用T''取代T中的T', 得到一個比T cost更低的binary search tree (矛盾)

T

$$\sum_{m=1}^{n}(depth_T(k_m)+1)\cdot p_m + \sum_{m=0}^{n}(depth_T(d_m)+1)\cdot q_m = E_{ij} + C$$

$$E'_{ij} + C < E_{ij} + C$$

矛盾!



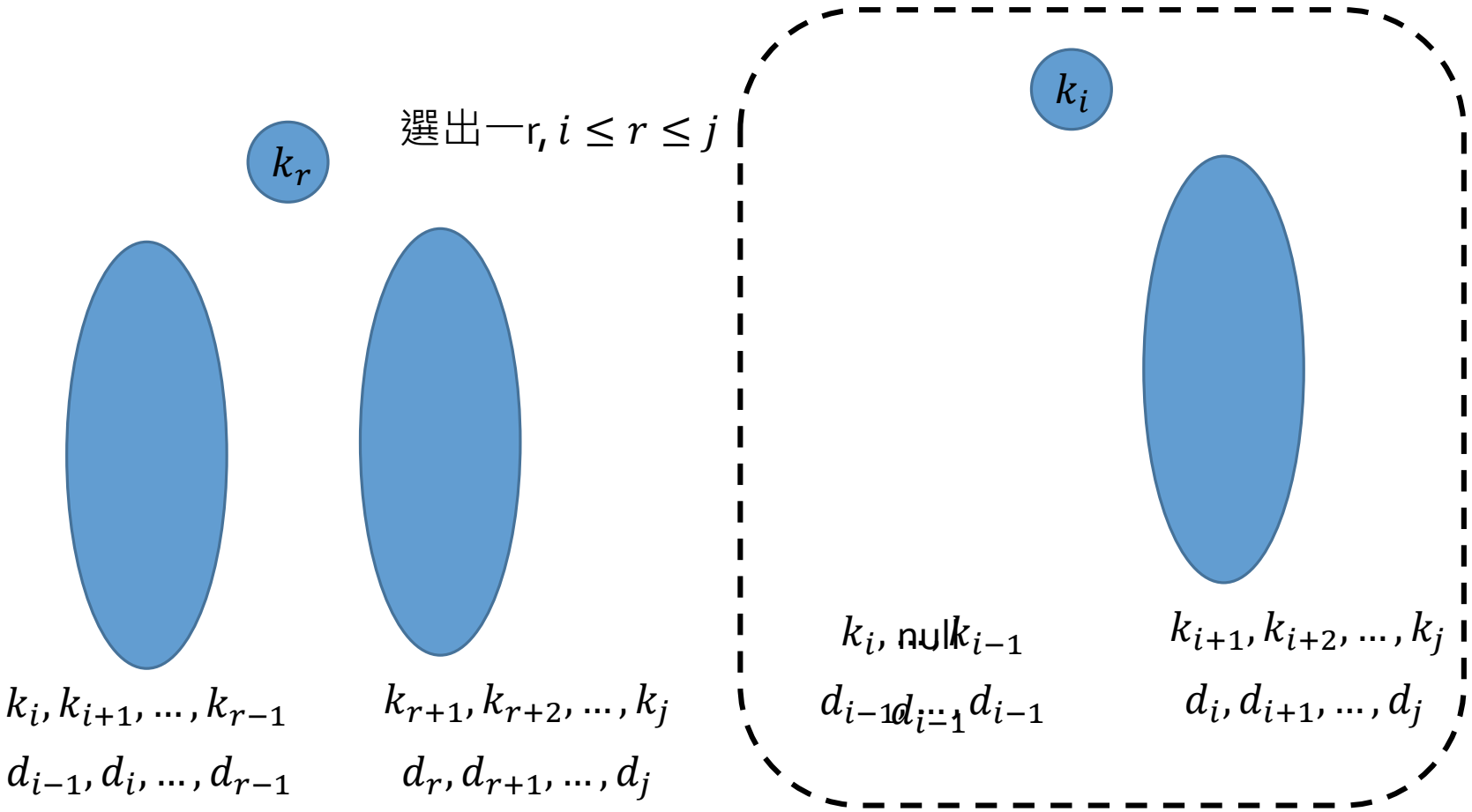
$k_i, k_{i+1}, \ldots, k_j$

$$E_{ij} = \sum_{m=i}^{j}(depth_T(k_m)+1)\cdot p_m + \sum_{m=i-1}^{j}(depth_T(d_m)+1)\cdot q_m$$

$E'_{ij} < \mathrm{E_{ij}}$

Dynamic Programming出招

- 如何用小問題的答案組出大問題的答案?

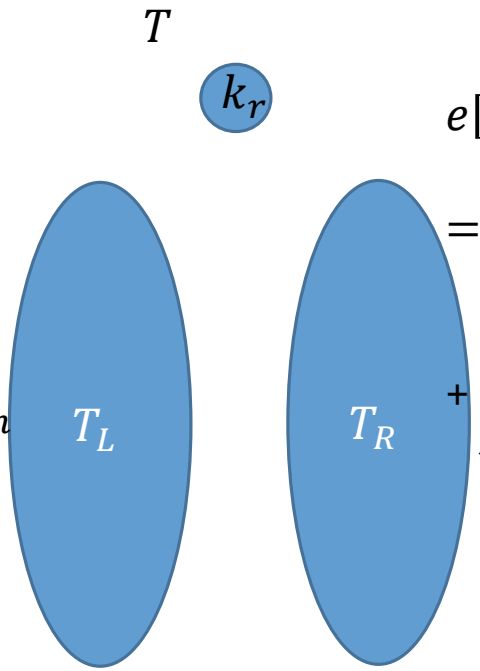


Dynamic Programming出招

2. 列出遞迴式子 (表示花費)

$$e[i, r-1] = \sum_{m=i}^{r-1} (depth_{T_L}(k_m) + 1) \cdot p_m + \sum_{m=i-1}^{r-1} (depth_{T_L}(d_m) + 1) \cdot q_m$$

$$e[r+1, j] = \sum_{m=r+1}^{j} (depth_{T_R}(k_m) + 1) \cdot p_m + \sum_{m=r+1}^{j} (depth_{T_R}(d_m) + 1) \cdot q_m$$



$$depth_T(\cdot) = depth_{T_L}(\cdot) + 1 = depth_{T_R}(\cdot) + 1$$

- $w(i,j) = \sum_{l=i}^{j} p_l + \sum_{l=i-1}^{j} q_l$

包含$k_i, \ldots, k_j$的subtree
所發生的機率

- $e[i,j] = p_r + \left(e[i,r-1] + w(i,r-$

條件不同, 使用的subproblem不同

,if $j = i-1$

,if $j \neq i-1$

r有多種選擇

Dynamic Programming出招

3. 計算花費

- 填表: e & w
  - e[i,j]: i=1 to n+1, j=0 to n
  - w[i,j]: i=1 to n+1, j=0 to n
- 為什麼w要填表? 不然計算每個e[i,j]都需要做$\Theta(j-i)$次加法

$$w[i,j] = \begin{cases} q_{i-1} & \text{,if } j = i-1 \\ w[i,j-1] + p_j + q_j & \text{,if } j \neq i-1 \end{cases}$$

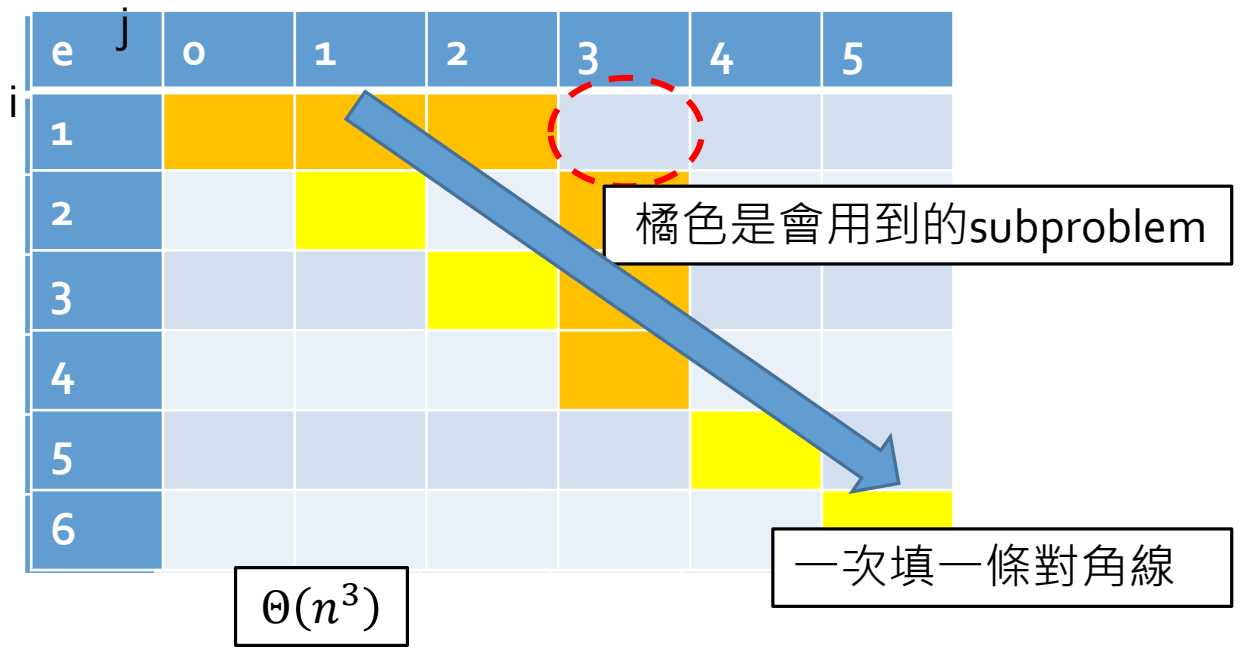
w	0	1	2	3	4	5
1						
2						
3						
4						
5						
6						

i

j

$\Theta(n^2)$

- e的填表順序



大家來練習

i	0	1	2	3	4	5
p_i		0.15	0.10	0.05	0.10	0.20
q_i	0.05	0.10	0.05	0.05	0.05	0.10

$$w[i,j] = \begin{cases} q_{i-1} & \text{,if } j = i-1 \\ w[i,j-1] + p_j + q_j & \text{,if } j \neq i-1 \end{cases}$$

$$e[i,j] = \begin{cases} q_{i-1} & \text{,if } j = i-1 \\ \min_{i \le r \le j} \{e[i,r-1] + e[r+1,j] + w(i,j)\} & \text{,if } j \neq i-1 \end{cases}$$

```
Optimal_BST(p,q,n)
let e[1..n+1,0..n],w[1..n+1,0..n],and
root[1..n,1..n] be new tables
for i=1 to n+1
      e[i,i-1]=q_{i-1}
      w[i,i-1]=q_{i-1}
for l=1 to n
      for i=1 to n-l+1
            j=i+l-1
            e[i,j]=∞
            w[i,j]=w[i,j-1]+p_j+q_j
            for r=i to j
                  t=e[i,r-1]+e[r+1,j]+w[i,j]
                  if t<e[i,j]
                        e[i,j]=t
                        root[i,j]=r
return e and root
```

e紀錄expected cost, root紀錄選擇結果

邊界起始值

填表: 兩層迴圈, 對角線順序

$\Theta(n^3)$

Dynamic Programming出招

4. 印出Optimal Binary Search Tree結果

- 自己回家研究一下☺ 15.5-1 on p. 403