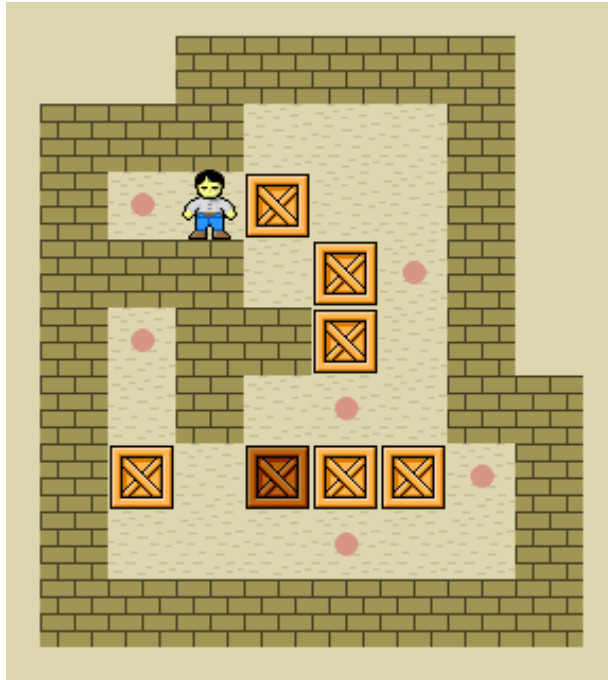


3_簡單的倉庫番 (Simple Sokoban)

(10分)

題目敘述

倉庫番是一款經典的電子遊戲，玩家需要使用上下左右操控畫面中的人物去推箱子，將所有箱子移動到目標格子，即可通關。



喵喵遊俠覺得倉庫番太難了，他決定把地圖中的人物拿掉，並重新設計遊戲。新設計的規則是：給你一個地圖，地圖上有牆壁與數個箱子與數個目標格子，箱子與目標格子的數量一樣。每一次操作，可以直接移動任一個箱子到上下左右的位置，箱子不能移動到牆壁上也不能超出地圖範圍且任意時刻不能有兩個箱子在同一個位置上。玩家只要移動所有箱子使得所有箱子都在目標格子上即可通關。

現在喵喵遊俠根據這個規則設計了好幾個地圖，但不確定每個地圖是否都存在通關的方法。請幫喵喵遊俠判斷地圖是否能順利通關。

輸入格式

第一行輸入一個數字 T 代表接下來有 T 個地圖需要判斷。

對於每一個地圖，一開始輸入兩個數字 N, M 代表地圖大小是 $N \times M$ 。

接著輸入 N 行，每一行包含 M 個字元，第 i 行的第 j 個字元代表地圖中位置 (i, j) 是什麼，有四種符號：

- #：牆壁
- A：箱子
- B：目標格子
- .：空的格子

輸出格式

對於每個地圖，輸出一行，若存在通關的方法就輸出 YES，否則輸出 NO。

資料範圍

- $1 \leq T \leq 10$
- $1 \leq N, M \leq 1000$
- 保證地圖中至少有一個箱子
- 箱子的數量等於目標格子的數量

測試範例

輸入範例 1

```
1
5 5
##.B#
..AA#
B####
#A..B
...##
```

輸出範例 1

```
YES
```

輸入範例 2

```
2
3 5
A.#.B
..#..
.....
3 5
A.#.B
..#..
..#..
```

輸出範例 2

```
YES
NO
```

輸入範例 3

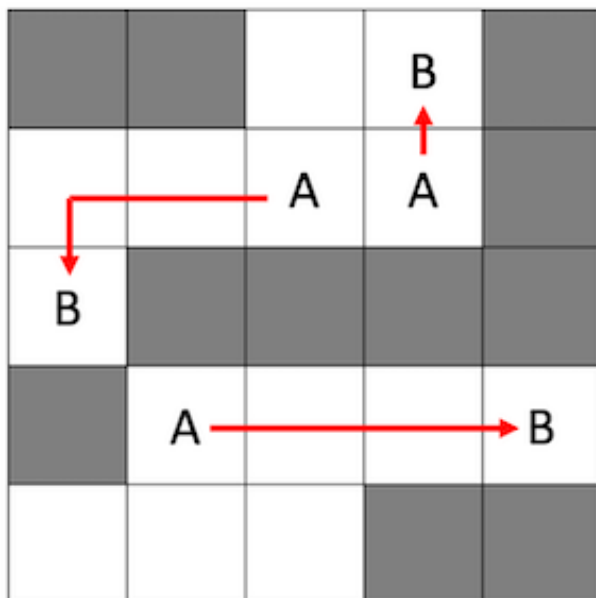
```
1
5 4
#.B.
.#..
A.#.
.#B.
.##A
```

輸出範例 3

NO

範例說明

- 以下是第一筆範例中一組可行的通關方法



- 第二筆範例中的第一個地圖可以輕易找出移動方法，至於第二個地圖，因為加上一個牆壁後箱子就無法移動到目標格，因此無通關方法。
- 第三筆範例中，左邊的箱子因為牆壁的阻擋無法移動到目標格子上，因此無通關方法。