算法设计与分析第二次作业动态规划

6-1 Programming Contest (30 分)

Bob will participate in a programming contest. There are altogether `n` problems in the contest. Unlike in PAT (Programming Ability Test), in a programming contest one can not obtain partial scores. For problem `i`, Bob will need `time[i]` to solve it and obtains the corresponding `score[i]`, or he may choose not to solve it at all. Bob will be happy when he obtains a total score no less than `happy_score`. You are supposed to find the minimum time needed for Bob to be happy. The function `need_time` must return the minimum time, or `-1` if it is impossible for Bob to obtain a score no less than `happy_score`.

Format of function:

```
int need_time(const int time[], const int score[], int happy_score, int n);
```

Here `n` (**1≤**`n`**≤** `MAXN`) is the number of problems; `happy_score` (**1≤**`happy_score` **≤** `MAXS`) is the minimum score for Bob to be happy; `time[]` is the array to store `time[i]` (**1****≤**`time[i]`**≤10****0**) which is the time to solve problem `i`; `score[]` is the array to store `score[i]` (**1****≤**`score[i]`**≤10****0**) which is the score Bob gets for solving problem `i`.

Sample program of judge:

```
#include <stdio.h>

#define MAXN 10
```

```c
#define MAXS 1000

int need_time(const int time[], const int score[], int happy_score, int n);

int main() {
  int n, i, happy_score;
  int time[MAXN], score[MAXN];
  scanf("%d %d", &n, &happy_score);
  for (i = 0; i < n; ++i)
      scanf("%d", &time[i]);
  for (i = 0; i < n; ++i)
      scanf("%d", &score[i]);
  printf("%d\n", need_time(time, score, happy_score, n));
  return 0;
}

/* Your function will be put here */
```