

分组背包与环形 DP

Hatsune Miku

GRYZ

2018 年 4 月 19 日

自我介绍

自我介绍

全机房最弱

自我介绍

自我介绍

全机房最弱

不会讲课

自我介绍

自我介绍

全机房最弱

不会讲课

zxl 说不能只让 txt 同学听懂, skh 同学也要听懂

自我介绍

自我介绍

全机房最弱

不会讲课

zxl 说不能只让 txt 同学听懂, skh 同学也要听懂

其他同学就.....

自我介绍

自我介绍

全机房最弱

不会讲课

zxl 说不能只让 txt 同学听懂, skh 同学也要听懂

其他同学就....

关于我的讲课风格

自我介绍

自我介绍

全机房最弱

不会讲课

zxl 说不能只让 txt 同学听懂, skh 同学也要听懂

其他同学就....

关于我的讲课风格

很多人吐槽我的讲课风格很差?

自我介绍

自我介绍

全机房最弱

不会讲课

zxl 说不能只让 txt 同学听懂, skh 同学也要听懂

其他同学就....

关于我的讲课风格

很多人吐槽我的讲课风格很差?

可能确实是那样

自我介绍

自我介绍

全机房最弱

不会讲课

zxl 说不能只让 txt 同学听懂, skh 同学也要听懂

其他同学就....

关于我的讲课风格

很多人吐槽我的讲课风格很差?

可能确实是那样

没办法

自我介绍

自我介绍

全机房最弱

不会讲课

zxl 说不能只让 txt 同学听懂, skh 同学也要听懂

其他同学就....

关于我的讲课风格

很多人吐槽我的讲课风格很差?

可能确实是那样

没办法

毕竟全机房最弱

关于这节课

关于 PPT 与讲课的问题

关于这节课

关于 PPT 与讲课的问题

这节课是我草草准备了两三天的产物，PPT 也非常丑。

关于这节课

关于 PPT 与讲课的问题

这节课是我草草准备了两三天的产物，PPT 也非常丑。
这节课还是比较有难度的，我可能讲着讲着就讲不下去了。

关于这节课

关于 PPT 与讲课的问题

这节课是我草草准备了两三天的产物，PPT 也非常丑。

这节课还是比较有难度的，我可能讲着讲着就讲不下去了。

凉了凉了，不讲了（逃

关于这节课

关于 PPT 与讲课的问题

这节课是我草草准备了两三天的产物，PPT 也非常丑。
这节课还是比较有难度的，我可能讲着讲着就讲不下去了。
凉了凉了，不讲了（逃

主要内容

关于这节课

关于 PPT 与讲课的问题

这节课是我草草准备了两三天的产物，PPT 也非常丑。
这节课还是比较有难度的，我可能讲着讲着就讲不下去了。
凉了凉了，不讲了（逃

主要内容

据说在座各位水平很高，尤其是 txt 和 skh 同学，水平已经吊打我了？

关于这节课

关于 PPT 与讲课的问题

这节课是我草草准备了两三天的产物，PPT 也非常丑。
这节课还是比较有难度的，我可能讲着讲着就讲不下去了。
凉了凉了，不讲了（逃

主要内容

据说在座各位水平很高，尤其是 txt 和 skh 同学，水平已经吊打我了？
（我还是不讲了吧，毕竟你们都会了

关于这节课

关于 PPT 与讲课的问题

这节课是我草草准备了两三天的产物，PPT 也非常丑。
这节课还是比较有难度的，我可能讲着讲着就讲不下去了。
凉了凉了，不讲了（逃

主要内容

据说在座各位水平很高，尤其是 txt 和 skh 同学，水平已经吊打我了？
（我还是不讲了把，毕竟你们都会了还是要讲一讲的。课主要分两个部分，其实都是之前的 dalao 讲过的内容：

关于这节课

关于 PPT 与讲课的问题

这节课是我草草准备了两三天的产物，PPT 也非常丑。
这节课还是比较有难度的，我可能讲着讲着就讲不下去了。
凉了凉了，不讲了（逃

主要内容

据说在座各位水平很高，尤其是 txt 和 skh 同学，水平已经吊打我了？
（我还是不讲了吧，毕竟你们都会了还是要讲一讲的。课主要分两个部分，其实都是之前的 dalao 讲过的内容：

- ① 分组背包
- ② 环形 DP

前置知识

很多人说，我的课要求前置知识太多，讲着讲着就讲到别的地方去了。

前置知识

很多人说，我的课要求前置知识太多，讲着讲着就讲到别的地方去了。
那是你前置知识没学好啊喂！

前置知识

很多人说，我的课要求前置知识太多，讲着讲着就讲到别的地方去了。

那是你前置知识没学好啊喂！

先说一下前置知识，都是以前学过的东西：

前置知识

很多人说，我的课要求前置知识太多，讲着讲着就讲到别的地方去了。

那是你前置知识没学好啊喂!

先说一下前置知识，都是以前学过的东西：

① 搜索

前置知识

很多人说，我的课要求前置知识太多，讲着讲着就讲到别的地方去了。

那是你前置知识没学好啊喂!

先说一下前置知识，都是以前学过的东西：

① 搜索

- DFS

前置知识

很多人说，我的课要求前置知识太多，讲着讲着就讲到别的地方去了。

那是你前置知识没学好啊喂!

先说一下前置知识，都是以前学过的东西：

① 搜索

- DFS
- BFS

前置知识

很多人说，我的课要求前置知识太多，讲着讲着就讲到别的地方去了。

那是你前置知识没学好啊喂！

先说一下前置知识，都是以前学过的东西：

① 搜索

- DFS
- BFS
- * 记忆化搜索

前置知识

很多人说，我的课要求前置知识太多，讲着讲着就讲到别的地方去了。

那是你前置知识没学好啊喂！

先说一下前置知识，都是以前学过的东西：

① 搜索

- DFS
- BFS
- * 记忆化搜索

② DP

前置知识

很多人说，我的课要求前置知识太多，讲着讲着就讲到别的地方去了。

那是你前置知识没学好啊喂!

先说一下前置知识，都是以前学过的东西：

① 搜索

- DFS
- BFS
- * 记忆化搜索

2 DP

- DP 的基本知识
- 背包

前置知识

很多人说，我的课要求前置知识太多，讲着讲着就讲到别的地方去了。

那是你前置知识没学好啊喂!

先说一下前置知识，都是以前学过的东西：

① 搜索

- DFS
- BFS
- * 记忆化搜索

② DP

- DP 的基本知识
- 背包
 - 01 背包

前置知识

很多人说，我的课要求前置知识太多，讲着讲着就讲到别的地方去了。

那是你前置知识没学好啊喂!

先说一下前置知识，都是以前学过的东西：

① 搜索

- DFS
- BFS
- * 记忆化搜索

② DP

- DP 的基本知识
- 背包
 - 01 背包
 - 完全背包

前置知识

很多人说，我的课要求前置知识太多，讲着讲着就讲到别的地方去了。

那是你前置知识没学好啊喂!

先说一下前置知识，都是以前学过的东西：

① 搜索

- DFS
- BFS
- * 记忆化搜索

② DP

- DP 的基本知识
- 背包
 - 01 背包
 - 完全背包
 - 多重背包

前置知识

很多人说，我的课要求前置知识太多，讲着讲着就讲到别的地方去了。

那是你前置知识没学好啊喂!

先说一下前置知识，都是以前学过的东西：

① 搜索

- DFS
- BFS
- * 记忆化搜索

② DP

- DP 的基本知识
- 背包
 - 01 背包
 - 完全背包
 - 多重背包
 - 混合背包

前置知识

很多人说，我的课要求前置知识太多，讲着讲着就讲到别的地方去了。

那是你前置知识没学好啊喂!

先说一下前置知识，都是以前学过的东西：

① 搜索

- DFS
- BFS
- * 记忆化搜索

② DP

- DP 的基本知识
- 背包
 - 01 背包
 - 完全背包
 - 多重背包
 - 混合背包
- 其他 DP 类型

前置知识

很多人说，我的课要求前置知识太多，讲着讲着就讲到别的地方去了。

那是你前置知识没学好啊喂!

先说一下前置知识，都是以前学过的东西：

① 搜索

- DFS
- BFS
- * 记忆化搜索

② DP

- DP 的基本知识
- 背包
 - 01 背包
 - 完全背包
 - 多重背包
 - 混合背包
- 其他 DP 类型
 - 序列型 DP

前置知识

很多人说，我的课要求前置知识太多，讲着讲着就讲到别的地方去了。

那是你前置知识没学好啊喂!

先说一下前置知识，都是以前学过的东西：

① 搜索

- DFS
- BFS
- * 记忆化搜索

② DP

- DP 的基本知识
- 背包
 - 01 背包
 - 完全背包
 - 多重背包
 - 混合背包
- 其他 DP 类型
 - 序列型 DP
 - 区间 DP

啥?

妈妈妈妈，我要回家。

啥?

妈妈妈妈，我要回家。

垃圾竞赛，费我时间，耗我钱财，拿不到一等奖，照样完蛋!

啥?

妈妈妈妈，我要回家。

垃圾竞赛，费我时间，耗我钱财，拿不到一等奖，照样完蛋！

别怕 qwq，之前不是讲过吗？

啥?

妈妈妈妈，我要回家。

垃圾竞赛，费我时间，耗我钱财，拿不到一等奖，照样完蛋！

别怕 qwq，之前不是讲过吗？

我就假装你们都会了这些东西

简单介绍

这一部分，我将会用 5 道左右的例题来介绍分组背包这一类问题的基本解法。题目难度不定，有些可能还需要一些技巧才能完成。

简单介绍

这一部分，我将会用 5 道左右的例题来介绍分组背包这一类问题的基本解法。题目难度不定，有些可能还需要一些技巧才能完成。

首先让我们看看什么是分组背包，形式的定义一下分组背包：

简单介绍

这一部分，我将会用 5 道左右的例题来介绍分组背包这一类问题的基本解法。题目难度不定，有些可能还需要一些技巧才能完成。

首先让我们看看什么是分组背包，形式的定义一下分组背包：

分组背包的定义

给你若干个物品，这些物品可以分为若干组，每件物品都有一个体积和价值，你有一个背包，背包有一定容量，规定每一组物品只能选择一件（若干件也可以），问能获得的最大价值是多少。

啊，这简单……就加一层枚举物品所属的组就行了。

简单介绍

这一部分，我将会用 5 道左右的例题来介绍分组背包这一类问题的基本解法。题目难度不定，有些可能还需要一些技巧才能完成。

首先让我们看看什么是分组背包，形式的定义一下分组背包：

分组背包的定义

给你若干个物品，这些物品可以分为若干组，每件物品都有一个体积和价值，你有一个背包，背包有一定容量，规定每一组物品只能选择一件（若干件也可以），问能获得的最大价值是多少。

啊，这简单……就加一层枚举物品所属的组就行了。

回答正确。

简单介绍

这一部分，我将会用 5 道左右的例题来介绍分组背包这一类问题的基本解法。题目难度不定，有些可能还需要一些技巧才能完成。

首先让我们看看什么是分组背包，形式的定义一下分组背包：

分组背包的定义

给你若干个物品，这些物品可以分为若干组，每件物品都有一个体积和价值，你有一个背包，背包有一定容量，规定每一组物品只能选择一件（若干件也可以），问能获得的最大价值是多少。

啊，这简单……就加一层枚举物品所属的组就行了。

回答正确。

但，

简单介绍

这一部分，我将会用 5 道左右的例题来介绍分组背包这一类问题的基本解法。题目难度不定，有些可能还需要一些技巧才能完成。

首先让我们看看什么是分组背包，形式的定义一下分组背包：

分组背包的定义

给你若干个物品，这些物品可以分为若干组，每件物品都有一个体积和价值，你有一个背包，背包有一定容量，规定每一组物品只能选择一件（若干件也可以），问能获得的最大价值是多少。

啊，这简单……就加一层枚举物品所属的组就行了。

回答正确。

但，问题真的这么简单吗？

两份代码

我们假设，每组物品只能选一件（01 背包），给出两份代码，请指出哪一份代码是正确的：

两份代码

我们假设，每组物品只能选一件（01 背包），给出两份代码，请指出哪一份代码是正确的：

代码 1

```
for each group k:
    for j in backpack.size to 0:
        for each element i in group k:
            f[j]=max{f[j],f[j-c[i]]+w[i]}
```

两份代码

我们假设，每组物品只能选一件（01 背包），给出两份代码，请指出哪一份代码是正确的：

代码 1

```
for each group k:
    for j in backpack.size to 0:
        for each element i in group k:
            f[j]=max{f[j],f[j-c[i]]+w[i]}
```

代码 2

```
for each group k:
    for each element i in group k:
        for j in backpack.size to 0:
            f[j]=max{f[j],f[j-c[i]]+w[i]}
```

关于这节课的一些事情
○○

前置知识
○○

分组背包问题
○○●

例题
○○○○○○○○

环形 DP
○○

例题
○○○○○○○○○○○○○○○○

致谢
○

思考一下

思考一下

答案

上一张 PPT 中代码 1 是正确的。

为什么？

思考一下代码 1 的含义，本身是在背包容量一定的前提下，枚举每一组的每一个物品；代码 2 则调换了枚举物品和枚举背包体积的顺序，先枚举每一件物品，然后枚举背包体积，有什么发现？

思考一下

答案

上一张 PPT 中代码 1 是正确的。

为什么？

思考一下代码 1 的含义，本身是在背包容量一定的前提下，枚举每一组的每一个物品；代码 2 则调换了枚举物品和枚举背包体积的顺序，先枚举每一件物品，然后枚举背包体积，有什么发现？

显然代码 1 保证了在背包容量一定的前提下，每一个容量对应一个最优解。代码 2 则是保证了每一件物品的体积一定的前提下，每一个容量对应一个最优解。

思考一下

答案

上一张 PPT 中代码 1 是正确的。

为什么？

思考一下代码 1 的含义，本身是在背包容量一定的前提下，枚举每一组的每一个物品；代码 2 则调换了枚举物品和枚举背包体积的顺序，先枚举每一件物品，然后枚举背包体积，有什么发现？

显然代码 1 保证了在背包容量一定的前提下，每一个容量对应一个最优解。代码 2 则是保证了每一件物品的体积一定的前提下，每一个容量对应一个最优解。

明白了吗？就是说，代码 1 保证每个背包容量只能对应选择 1 件物品，而代码 2 不能保证在每一组物品只选择 1 件，不符合分组背包的定义，这就凉了。

思考一下

答案

上一张 PPT 中代码 1 是正确的。

为什么？

思考一下代码 1 的含义，本身是在背包容量一定的前提下，枚举每一组的每一个物品；代码 2 则调换了枚举物品和枚举背包体积的顺序，先枚举每一件物品，然后枚举背包体积，有什么发现？

显然代码 1 保证了在背包容量一定的前提下，每一个容量对应一个最优解。代码 2 则是保证了每一件物品的体积一定的前提下，每一个容量对应一个最优解。

明白了吗？就是说，代码 1 保证每个背包容量只能对应选择 1 件物品，而代码 2 不能保证在每一组物品只选择 1 件，不符合分组背包的定义，这就凉了。

当然你也可以套上其他的背包模板，使这个简单的分组 01 背包变得复杂一些。

几个题目

让我们来看几道例题巩固一下:

- 洛谷 P1757 通天之分组背包
- HDU 1712 ACboy needs your help
- POJ 3046 Ant Counting
- HDU 4341 Gold Miner
- HDU 3033 I love sneakers!

◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡ ≡ ≡ ≡ ↺ 🔍 ↻

关于这节课的一些事情
○○

前置知识
○○

分组背包问题
○○○

例题
●○○○○○○○

环形 DP
○○

例题
○○○○○○○○○○○○○○○○○○

致谢
○

dalao 讲题时间

洛谷 P1757 通天之分组背包

题目描述

自 01 背包问世之后，小 A 对此深感兴趣。一天，小 A 去远游，却发现他的背包不同于 01 背包，他的物品大致可分为 k 组，每组中的物品相互冲突，现在，他想知道最大的利用价值是多少。

输入格式

两个数 m, n ，表示一共有 n 件物品，总重量为 m

接下来 n 行，每行 3 个数 a_i, b_i, c_i ，表示第 i 件物品的重量，利用价值，所属组数

输出格式

一个数，最大的利用价值

$1 \leq m \leq 1000, 1 \leq n \leq 1000$ ，组数 $t \leq 100$

分组背包模板题。

分组背包模板题。上代码？

分组背包模板题。上代码？

```

#include<cstdio>
#include<algorithm>
using namespace std;
int f[1005],ai[105][1000],ci[1005],wi[1005];
int main(){
    scanf("%d%d",&m,&n);
    for(int i=1; i<=n; i++){
        scanf("%d%d%d",&wi[i],&ci[i],&j);
        ai[j][++ai[j][0]]=i;
    }
    for(int i=1; ai[i][0]!=0&&i<=100; i++)
        for(int j=m; j>=1; j--)
            for(int t=1; t<=ai[i][0]; t++)
                if(j>=wi[ai[i][t]])
                    f[j]=max(f[j],f[j-wi[ai[i][t]]]+ci[ai[i][t]]);
    printf("%d\n",f[m]);
    return 0;
}

```

HDU 1712 ACboy needs your help

题目描述

ACboy 这个学期有 N 门课要学习，他计划最多学习 M 天，每门课的学习效果与他的学习时长有关。如何安排才能使学习效果最好？

输入格式

多组测试数据，每组测试数据由两个正整数 N, M 组成，接下来是一个矩阵 $A[i][j]$ ， $A[i][j]$ 表示如果 ACboy 在第 i 门课上花费 j 天会获得的学习效果。
 $N = 0, M = 0$ 结束输入。

输出格式

对于每组数据，输出一行表示 ACboy 获得的最大学习效果。
 $1 \leq N, M \leq 100$

还是分组背包的裸题。

还是分组背包的裸题。
或者可以按照泛化物品来做（太明显了）。

还是分组背包的裸题。
或者可以按照泛化物品来做（太明显了）。
这样我们大概就顺便得到泛化物品的做法了。

还是分组背包的裸题。

或者可以按照泛化物品来做（太明显了）。

这样我们大概就顺便得到泛化物品的做法了。

什么，你不知道什么是泛化物品？

泛化物品的定义

在背包问题中，物品没有一定的体积及对应的价值，而是价值随分配给它的体积变化而变化，这样的物品叫做泛化物品。

更严格的定义，在背包容量为 V 的背包问题中，泛化物品是一个定义域为 \mathbf{N}_+ 中的函数 h ，当分配给它的费用为 v 时，能得到的价值就是 $h(v)$ 。

还是分组背包的裸题。
或者可以按照泛化物品来做（太明显了）。
这样我们大概就顺便得到泛化物品的做法了。
什么，你不知道什么是泛化物品？

泛化物品的定义

在背包问题中，物品没有一定的体积及对应的价值，而是价值随分配给它的体积变化而变化，这样的物品叫做泛化物品。

更严格的定义，在背包容量为 V 的背包问题中，泛化物品是一个定义域为 \mathbf{N}_+ 中的函数 h ，当分配给它的费用为 v 时，能得到的价值就是 $h(v)$ 。

体会一下这两种背包问题之间的关系。

还是分组背包的裸题。
或者可以按照泛化物品来做（太明显了）。
这样我们大概就顺便得到泛化物品的做法了。
什么，你不知道什么是泛化物品？

泛化物品的定义

在背包问题中，物品没有一定的体积及对应的价值，而是价值随分配给它的体积变化而变化，这样的物品叫做泛化物品。

更严格的定义，在背包容量为 V 的背包问题中，泛化物品是一个定义域为 \mathbf{N}_+ 中的函数 h ，当分配给它的费用为 v 时，能得到的价值就是 $h(v)$ 。

体会一下这两种背包问题之间的关系。
触类旁通，泛化物品也能做了。是不是感觉自己很棒棒啊？

POJ 3046 Ant Counting

题意

有 T 个蚂蚁家族，编号 $1 \sim T$ ，每个家族中蚂蚁有 $N[i]$ 只。用这些蚂蚁家族中的蚂蚁组成一个蚂蚁序列，问长度在 $S \sim B$ 之间的不同蚂蚁序列有多少种。

$1 \leq T, N[i] \leq 100$

POJ 3046 Ant Counting

题意

有 T 个蚂蚁家族，编号 $1 \sim T$ ，每个家族中蚂蚁有 $N[i]$ 只。用这些蚂蚁家族中的蚂蚁组成一个蚂蚁序列，问长度在 $S \sim B$ 之间的不同蚂蚁序列有多少种。

$1 \leq T, N[i] \leq 100$

怎么有些不一样？

开始有些难度了？不是模板题了？只能枚举子集了？没有思路了？考虑几分钟？

POJ 3046 Ant Counting

题意

有 T 个蚂蚁家族，编号 $1 \sim T$ ，每个家族中蚂蚁有 $N[i]$ 只。用这些蚂蚁家族中的蚂蚁组成一个蚂蚁序列，问长度在 $S \sim B$ 之间的不同蚂蚁序列有多少种。

$1 \leq T, N[i] \leq 100$

怎么有些不一样？

开始有些难度了？不是模板题了？只能枚举子集了？没有思路了？考虑几分钟？

题解

我们把每个蚂蚁家族看做一个分组，每一只蚂蚁看做一个物品，在每个分组里有 $N[i]$ 种选择，每种选择对答案的贡献为 1。然后就是分组背包了。

POJ 3046 Ant Counting

题意

有 T 个蚂蚁家族，编号 $1 \sim T$ ，每个家族中蚂蚁有 $N[i]$ 只。用这些蚂蚁家族中的蚂蚁组成一个蚂蚁序列，问长度在 $S \sim B$ 之间的不同蚂蚁序列有多少种。

$1 \leq T, N[i] \leq 100$

怎么有些不一样？

开始有些难度了？不是模板题了？只能枚举子集了？没有思路了？考虑几分钟？

题解

我们把每个蚂蚁家族看做一个分组，每一只蚂蚁看做一个物品，在每个分组里有 $N[i]$ 种选择，每种选择对答案的贡献为 1。然后就是分组背包了。

最后，根据题意，答案等于所有长度在区间 $[S, B]$ 的蚂蚁序列的种类数的和，即

$$ans = \sum_{i=S}^B dp[i].$$

POJ 3046 Ant Counting

题意

有 T 个蚂蚁家族，编号 $1 \sim T$ ，每个家族中蚂蚁有 $N[i]$ 只。用这些蚂蚁家族中的蚂蚁组成一个蚂蚁序列，问长度在 $S \sim B$ 之间的不同蚂蚁序列有多少种。

$1 \leq T, N[i] \leq 100$

怎么有些不一样？

开始有些难度了？不是模板题了？只能枚举子集了？没有思路了？考虑几分钟？

题解

我们把每个蚂蚁家族看做一个分组，每一只蚂蚁看做一个物品，在每个分组里有 $N[i]$ 种选择，每种选择对答案的贡献为 1。然后就是分组背包了。

最后，根据题意，答案等于所有长度在区间 $[S, B]$ 的蚂蚁序列的种类数的和，即

$$ans = \sum_{i=S}^B dp[i].$$

是不是很简单？

HDU 3033 I love sneakers!

题意

Iserlohn 喜欢收集鞋子。他一共有 M 块钱，一共有 N 种鞋子，分别属于 K 个牌子，第 i 双鞋子是 a_i 牌子的，标价为 b_i ，第 i 双鞋子都有一个价值 c_i 。他希望每个牌子的鞋子至少买一双，并且一双鞋不会买两次。问他能获得的最大价值是多少。

$1 \leq N \leq 100, 1 \leq M \leq 10000, 1 \leq K \leq 10, 1 \leq a \leq K, 0 \leq b, c < 100000$

HDU 3033 I love sneakers!

题意

Iserlohn 喜欢收集鞋子。他一共有 M 块钱，一共有 N 种鞋子，分别属于 K 个牌子，第 i 双鞋子是 a_i 牌子的，标价为 b_i ，第 i 双鞋子都有一个价值 c_i 。他希望每个牌子的鞋子至少买一双，并且一双鞋不会买两次。问他能获得的最大价值是多少。

$1 \leq N \leq 100, 1 \leq M \leq 10000, 1 \leq K \leq 10, 1 \leq a \leq K, 0 \leq b, c < 100000$

怎么又不一样了？

每组物品中至少要选一个，这怎么做啊？

HDU 3033 I love sneakers!

题意

Iserlohn 喜欢收集鞋子。他一共有 M 块钱，一共有 N 种鞋子，分别属于 K 个牌子，第 i 双鞋子是 a_i 牌子的，标价为 b_i ，第 i 双鞋子都有一个价值 c_i 。他希望每个牌子的鞋子至少买一双，并且一双鞋不会买两次。问他能获得的最大价值是多少。

$1 \leq N \leq 100, 1 \leq M \leq 10000, 1 \leq K \leq 10, 1 \leq a \leq K, 0 \leq b, c < 100000$

怎么又不一样了？

每组物品中至少要选一个，这怎么做啊？考虑特殊标记一下不符合题意的状态？

HDU 3033 I love sneakers!

题意

Iserlohn 喜欢收集鞋子。他一共有 M 块钱，一共有 N 种鞋子，分别属于 K 个牌子，第 i 双鞋子是 a_i 牌子的，标价为 b_i ，第 i 双鞋子都有一个价值 c_i 。他希望每个牌子的鞋子至少买一双，并且一双鞋不会买两次。问他能获得的最大价值是多少。

$1 \leq N \leq 100, 1 \leq M \leq 10000, 1 \leq K \leq 10, 1 \leq a \leq K, 0 \leq b, c < 100000$

怎么又不一样了？

每组物品中至少要选一个，这怎么做啊？考虑特殊标记一下不符合题意的状态？

题解

我们把 DP 数组初始化成 -1 表示不合法，转移的时候只从合法的状态转移。

HDU 3033 I love sneakers!

题意

Iserlohn 喜欢收集鞋子。他一共有 M 块钱，一共有 N 种鞋子，分别属于 K 个牌子，第 i 双鞋子是 a_i 牌子的，标价为 b_i ，第 i 双鞋子都有一个价值 c_i 。他希望每个牌子的鞋子至少买一双，并且一双鞋不会买两次。问他能获得的最大价值是多少。

$1 \leq N \leq 100, 1 \leq M \leq 10000, 1 \leq K \leq 10, 1 \leq a \leq K, 0 \leq b, c < 100000$

怎么又不一样了？

每组物品中至少要选一个，这怎么做啊？考虑特殊标记一下不符合题意的状态？

题解

我们把 DP 数组初始化成 -1 表示不合法，转移的时候只从合法的状态转移。对于 DP 数组的第 0 行，为了能够计算第 1 行，不能初始化为 -1 ，很明显应该初始化为 0。否则连初始状态都无法转移过去，那就凉了。

HDU 3033 I love sneakers!

题意

Iserlohn 喜欢收集鞋子。他一共有 M 块钱，一共有 N 种鞋子，分别属于 K 个牌子，第 i 双鞋子是 a_i 牌子的，标价为 b_i ，第 i 双鞋子都有一个价值 c_i 。他希望每个牌子的鞋子至少买一双，并且一双鞋不会买两次。问他能获得的最大价值是多少。

$1 \leq N \leq 100, 1 \leq M \leq 10000, 1 \leq K \leq 10, 1 \leq a \leq K, 0 \leq b, c < 100000$

怎么又不一样了？

每组物品中至少要选一个，这怎么做啊？考虑特殊标记一下不符合题意的状态？

题解

我们把 DP 数组初始化成 -1 表示不合法，转移的时候只从合法的状态转移。

对于 DP 数组的第 0 行，为了能够计算第 1 行，不能初始化为 -1 ，很明显应该初始化为 0。否则连初始状态都无法转移过去，那就凉了。

剩下的就是个普通的分组背包。

简单介绍

这一部分，我还是会用 5 道左右的例题来介绍环形 DP 这一类问题的基本解法。题目难度还是不定，有些题目需要一些技巧（不是裸的环形 DP）才能完成。

简单介绍

这一部分，我还是会用 5 道左右的例题来介绍环形 DP 这一类问题的基本解法。题目难度还是不定，有些题目需要一些技巧（不是裸的环形 DP）才能完成。与背包问题不同的是，环形 DP 这一类问题没有什么固定的套路，也就是说

简单介绍

这一部分，我还是会用 5 道左右的例题来介绍环形 DP 这一类问题的基本解法。题目难度还是不定，有些题目需要一些技巧（不是裸的环形 DP）才能完成。与背包问题不同的是，环形 DP 这一类问题没有什么固定的套路，也就是说你的 DP 方程得手推了。

简单介绍

这一部分，我还是会用 5 道左右的例题来介绍环形 DP 这一类问题的基本解法。题目难度还是不定，有些题目需要一些技巧（不是裸的环形 DP）才能完成。与背包问题不同的是，环形 DP 这一类问题没有什么固定的套路，也就是说你的 DP 方程得手推了。
惨啊。

简单介绍

这一部分，我还是会用 5 道左右的例题来介绍环形 DP 这一类问题的基本解法。题目难度还是不定，有些题目需要一些技巧（不是裸的环形 DP）才能完成。

与背包问题不同的是，环形 DP 这一类问题没有什么固定的套路，也就是说你的 DP 方程得手推了。

惨啊。但是，还是有一定套路可循的。基本思想是拆环成链。就是把环上的 DP 问题转化为序列上的 DP 问题。

我们还是先形式化的给环形 DP 一个定义：

简单介绍

这一部分，我还是会用 5 道左右的例题来介绍环形 DP 这一类问题的基本解法。题目难度还是不定，有些题目需要一些技巧（不是裸的环形 DP）才能完成。

与背包问题不同的是，环形 DP 这一类问题没有什么固定的套路，也就是说你的 DP 方程得手推了。

惨啊。但是，还是有一定套路可循的。基本思想是拆环成链。就是把环上的 DP 问题转化为序列上的 DP 问题。

我们还是先形式化的给环形 DP 一个定义：

环形 DP 的定义

环形 DP 本身是序列型 DP 的一个特例，其特点是 DP 的对象是一个环，而不是一个序列。其基本思想就是拆环成链。

什么是拆环成链？

所谓拆环成链，就是把一个环拆成一条链。（相当于啥也没说）

什么是拆环成链？

所谓拆环成链，就是把一个环拆成一条链。（相当于啥也没说）
具体做法？dalao 来说一下？

什么是拆环成链？

所谓拆环成链，就是把一个环拆成一条链。（相当于啥也没说）

具体做法？dalao 来说一下？

其实就是把一个环从某一个点切一刀，断成一条链，然后把这条链复制一次。

听起来很靠谱的样子，有什么问题吗？

什么是拆环成链?

所谓拆环成链，就是把一个环拆成一条链。（相当于啥也没说）

具体做法？dalao 来说一下？

其实就是把一个环从某一个点切一刀，断成一条链，然后把这条链复制一次。

听起来很靠谱的样子，有什么问题吗？

注意范围啊亲，DP 的时候不要枚举超过这个环的长度的区间。

还要注意空间啊，如果复制这条链会 MLE 的话，就枚举每一个点，从每一个点切一刀，然后做一遍 DP，最后从这些 DP 的结果中取一个最优。

什么是拆环成链？

所谓拆环成链，就是把一个环拆成一条链。（相当于啥也没说）

具体做法？dalao 来说一下？

其实就是把一个环从某一个点切一刀，断成一条链，然后把这条链复制一次。

听起来很靠谱的样子，有什么问题吗？

注意范围啊亲，DP 的时候不要枚举超过这个环的长度的区间。

还要注意空间啊，如果复制这条链会 MLE 的话，就枚举每一个点，从每一个点切一刀，然后做一遍 DP，最后从这些 DP 的结果中取一个最优。

所以事实上我们有两种拆环成链的方式，这两种方式互有长短，要视情况使用。

什么是拆环成链？

所谓拆环成链，就是把一个环拆成一条链。（相当于啥也没说）

具体做法？dalao 来说一下？

其实就是把一个环从某一个点切一刀，断成一条链，然后把这条链复制一次。

听起来很靠谱的样子，有什么问题吗？

注意范围啊亲，DP 的时候不要枚举超过这个环的长度的区间。

还要注意空间啊，如果复制这条链会 MLE 的话，就枚举每一个点，从每一个点切一刀，然后做一遍 DP，最后从这些 DP 的结果中取一个最优。

所以事实上我们有两种拆环成链的方式，这两种方式互有长短，要视情况使用。事实上并没有什么新东西对吧，还是普通的序列型 DP 和区间 DP。

什么是拆环成链？

所谓拆环成链，就是把一个环拆成一条链。（相当于啥也没说）

具体做法？dalao 来说一下？

其实就是把一个环从某一个点切一刀，断成一条链，然后把这条链复制一次。

听起来很靠谱的样子，有什么问题吗？

注意范围啊亲，DP 的时候不要枚举超过这个环的长度的区间。

还要注意空间啊，如果复制这条链会 MLE 的话，就枚举每一个点，从每一个点切一刀，然后做一遍 DP，最后从这些 DP 的结果中取一个最优。

所以事实上我们有两种拆环成链的方式，这两种方式互有长短，要视情况使用。

事实上并没有什么新东西对吧，还是普通的序列型 DP 和区间 DP。

那还要你讲什么？

什么是拆环成链？

所谓拆环成链，就是把一个环拆成一条链。（相当于啥也没说）

具体做法？dalao 来说一下？

其实就是把一个环从某一个点切一刀，断成一条链，然后把这条链复制一次。

听起来很靠谱的样子，有什么问题吗？

注意范围啊亲，DP 的时候不要枚举超过这个环的长度的区间。

还要注意空间啊，如果复制这条链会 MLE 的话，就枚举每一个点，从每一个点切一刀，然后做一遍 DP，最后从这些 DP 的结果中取一个最优。

所以事实上我们有两种拆环成链的方式，这两种方式互有长短，要视情况使用。

事实上并没有什么新东西对吧，还是普通的序列型 DP 和区间 DP。

那还要你讲什么？

实际上，环形 DP 实现起来还是有很多细节需要注意的。

几个题目

让我们来看几道例题巩固一下:

- 洛谷 P1043 [NOIp2003 普及组第二题] 数字游戏
- 洛谷 P1880 [NOI1995 第二题] 石子合并
- 洛谷 P1063 [NOIp2006 提高组第一题] 能量项链
- Vijos 1451 圆环取数
- 洛谷 P1070 [NOIp2009 普及组第四题] 道路游戏

几个题目

让我们来看几道例题巩固一下:

- 洛谷 P1043 [NOIp2003 普及组第二题] 数字游戏
- 洛谷 P1880 [NOI1995 第二题] 石子合并
- 洛谷 P1063 [NOIp2006 提高组第一题] 能量项链
- Vijos 1451 圆环取数
- 洛谷 P1070 [NOIp2009 普及组第四题] 道路游戏

这次画风比较正常。

几个题目

让我们来看几道例题巩固一下：

- 洛谷 P1043 [NOIp2003 普及组第二题] 数字游戏
- 洛谷 P1880 [NOI1995 第二题] 石子合并
- 洛谷 P1063 [NOIp2006 提高组第一题] 能量项链
- Vijos 1451 圆环取数
- 洛谷 P1070 [NOIp2009 普及组第四题] 道路游戏

这次画风比较正常。

dalao 们肯定都已经 AK 了。请几位同学上来讲一讲？

dalao 讲题时间

洛谷 P1043 [NOIp2003 普及组第二题] 数字游戏

题目描述

丁丁正在玩一个游戏。游戏是这样的：在你面前有一圈整数（一共 N 个），你要按顺序将其分为 M 个部分，各部分内的数字相加，相加所得的 M 个结果对 10 取模后再相乘，最终得到一个数 k 。游戏的要求是使你所得的 k 最大或者最小。
特别值得注意的是，无论是负数还是正数，对 10 取模的结果均为非负值。请你编写程序帮他赢得这个游戏。

输入格式

输入文件第一行有两个整数， n ($1 \leq n \leq 50$) 和 m ($1 \leq m \leq 9$)。以下 n 行每行有个整数，其绝对值不大于 10^4 ，按顺序给出圈中的数字，首尾相接。

输出格式

输出有两行，各包含一个非负整数。第一行是你程序得到的最小值，第二行是最大值。

环形 DP 裸题。

环形 DP 裸题。看看代码？

环形 DP 裸题。看看代码？

```
#include<bits/stdc++.h>
using namespace std;
int n,m,a[55*2],s[55*2],f[55*2][55*2][12],g[55*2][55*2][12],mx=-1e9,mn=1e9;
inline int sum(int i,int j){return (s[j]-s[i-1]+10)%10;}
int main() {scanf("%d%d",&n,&m);
    for(int i=1; i<=n; i++) scanf("%d",&a[i]),a[i+n]=a[i]=(a[i]%10+10)%10;
    for(int i=1; i<=n*2; i++)
        s[i]=(s[i-1]+a[i]+10)%10,f[i][i][1]=g[i][i][1]=(a[i]+10)%10;
    for(int i=1; i<=n*2; i++) for(int j=i+1; j<=n*2&& j<=i+n-1; j++){
        f[i][j][1]=g[i][j][1]=sum(i,j);
        if(j-i+1==n&&m==1) mx=max(mx,f[i][j][1]),mn=min(mn,g[i][j][1]);
        for(int k=2; k<=min(m,j-i+1); k++) {f[i][j][k]=-1e9,g[i][j][k]=1e9;
            for(int t=1; t<=j-i; t++)
                f[i][j][k]=max(f[i][j][k],f[i][j-t][k-1]*sum(j-t+1,j)),
                g[i][j][k]=min(g[i][j][k],g[i][j-t][k-1]*sum(j-t+1,j));
            if(j-i+1==n&&k==m) mx=max(mx,f[i][j][k]),mn=min(mn,g[i][j][k]);
        }
    }
    printf("%d\n%d",mn,mx);
    return 0;
}
```

洛谷 P1880 [NOI1995 第二题] 石子合并

题目描述

在一个圆形操场的四周摆放 N 堆石子，现要将石子有次序地合并成一堆。规定每次只能选相邻的 2 堆合并成新的一堆，并将新的一堆的石子数，记为该次合并的得分。试设计出 1 个算法，计算出将 N 堆石子合并成 1 堆的最小得分和最大得分。

输入格式

第 1 行是正整数 N ，表示有 N 堆石子。第 2 行有 N 个数，分别表示每堆石子的个数。

输出格式

输出共 2 行，第 1 行为最小得分，第 2 行为最大得分。

洛谷 P1880 [NOI1995 第二题] 石子合并

题目描述

在一个圆形操场的四周摆放 N 堆石子，现要将石子有次序地合并成一堆。规定每次只能选相邻的 2 堆合并成新的一堆，并将新的一堆的石子数，记为该次合并的得分。试设计出 1 个算法，计算出将 N 堆石子合并成 1 堆的最小得分和最大得分。

输入格式

第 1 行是正整数 N ，表示有 N 堆石子。第 2 行有 N 个数，分别表示每堆石子的个数。

输出格式

输出共 2 行，第 1 行为最小得分，第 2 行为最大得分。

题解

还是套路，拆环成链后求前缀和，然后枚举断点 k ，剩下的就是普通的区间 DP 了。

洛谷 P1063 [NOIp2006 提高组第一题] 能量项链

题目描述

给你一串有 N 颗能量珠的项链，每颗能量珠都有一个头标记和尾标记和一个能量值，相邻的两颗能量珠可以聚合，如果前一颗能量珠的头标记为 m ，尾标记为 r ，后一颗能量珠的头标记为 r ，尾标记为 n ，则聚合后释放的能量为 $m \times r \times n$ ，新产生的珠子的头标记为 m ，尾标记为 n 。设计一个聚合顺序，使项链释放的能量最大。

输入格式

第一行是一个正整数 N ($4 \leq N \leq 100$)，表示项链上珠子的个数。第二行是 N 个用空格隔开的正整数，所有的数均不超过 1000。第 i 个数为第 i 颗珠子的头标记 ($1 \leq i \leq N$)，当 $1 \leq i < N$ 时，第 i 颗珠子的尾标记应该等于第 $i+1$ 颗珠子的头标记。第 N 颗珠子的尾标记应该等于第 1 颗珠子的头标记。

输出格式

只有一行，是一个正整数 E ($E \leq 2.1 \times 10^9$)，为一个最优聚合顺序所释放的总能量。

环形 DP 裸题。

环形 DP 裸题。
代码?

环形 DP 裸题。

代码？

```
#include<bits/stdc++.h>
using namespace std;
int n,e[300],s[300][300],maxn=-1;
int main(){
    cin>>n;
    for(int i=1;i<=n;i++){cin>>e[i];e[i+n]=e[i];}
    for(int i=2;i<2*n;i++){
        for(int j=i-1;i-j<n&& j>=1;j--){
            for(int k=j;k<i;k++){
                s[j][i]=max(s[j][i],s[j][k]+s[k+1][i]+e[j]*e[k+1]*e[i+1]);
                if(s[j][i]>maxn)maxn=s[j][i];
            }
        }
    }
    cout<<maxn;
    return 0;
}
```

Vijos 1451 圆环取数

题目描述

给一个被划分为 n 个格子的圆环，每个格子上有一个正整数。两个格子的距离为两个格子之间的格子数。环的圆心处固定有一个指针，开始指向了圆环上的某个格子。可以取下指针所指的那个格子里的数和与这个格子距离不大于 k 的格子的数，取一个数的代价即这个数的值。指针是可以转动的，每次转动可以将指针转向其相邻的格子，代价为圆环上剩下的数的最大值。求将所有数取完所有数的最小代价。

输入格式

输入的第 1 行有两个正整数 n 和 k ，描述了圆环上的格子数与取数的范围。
第 2 行有 n 个正整数，按顺时针方向描述了圆环上每个格子上的数，且指针一开始指向了第 1 个数字所在的格子。所有整数之间用一个空格隔开，且不超过 10000。

输出格式

输出仅包括 1 个整数，为取完所有数的最小代价。

环形 DP+RMQ 问题。题目中要求一个区间的最大值，那就要写 RMQ 了。
什么？不知道什么是 RMQ 问题？啊，就是区间最大值或区间最小值问题。不带修改的话，ST 表就行了。

环形 DP+RMQ 问题。题目中要求一个区间的最大值，那就要写 RMQ 了。
什么？不知道什么是 RMQ 问题？啊，就是区间最大值或区间最小值问题。不带修改的话，ST 表就行了。
不会的可以去学习一下，主要运用了倍增的思想（这种思想在 OI 里很常见）。代码：

环形 DP+RMQ 问题。题目中要求一个区间的最大值，那就要写 RMQ 了。

什么？不知道什么是 RMQ 问题？啊，就是区间最大值或区间最小值问题。不带修改的话，ST 表就行了。

不会的可以去学习一下，主要运用了倍增的思想（这种思想在 OI 里很常见）。代码：

```
void initRMQ(int n){//n 表示序列的长度
    for(int i=1; i<=n; i++) mx[i][0]=a[i];
    for(int j=1; j<=12; j++)
        for(int i=1; i+(1<<j)-1<=n; i++)
            mx[i][j]=max(mx[i][j-1],mx[i+(1<<(j-1))][j-1]);
}

int RMQ(int l,int r){//查询区间 [l,r] 的 RMQ
    if(l>r) return 0;
    int k=log(r-l+1)/log(2);
    return max(mx[l][k],mx[r-(1<<k)+1][k]);
}
```

环形 DP+RMQ 问题。题目中要求一个区间的最大值，那就要写 RMQ 了。

什么？不知道什么是 RMQ 问题？啊，就是区间最大值或区间最小值问题。不带修改的话，ST 表就行了。

不会的可以去学习一下，主要运用了倍增的思想（这种思想在 OI 里很常见）。代码：

```
void initRMQ(int n){//n 表示序列的长度
    for(int i=1; i<=n; i++) mx[i][0]=a[i];
    for(int j=1; j<=12; j++)
        for(int i=1; i+(1<<j)-1<=n; i++)
            mx[i][j]=max(mx[i][j-1],mx[i+(1<<(j-1))][j-1]);
}

int RMQ(int l,int r){//查询区间 [l,r] 的 RMQ
    if(l>r) return 0;
    int k=log(r-l+1)/log(2);
    return max(mx[l][k],mx[r-(1<<k)+1][k]);
}
```

先使用 `initRMQ(n)` 初始化，然后查询只需要调用 `RMQ(L,R)` 即可，预处理的时间复杂度是 $O(n \log n)$ ，查询是 $O(1)$ 。

解决这个问题之后就没什么问题了。剩下的就是一个正常的环形 DP 了。

洛谷 P1070 [NOIp2009 普及组第四题] 道路游戏

题目描述

小新正在玩一个简单的电脑游戏。

游戏中有一条环形马路，马路上有 n 个机器人工厂，两个相邻机器人工厂之间由一小段马路连接。小新以某个机器人工厂为起点，按顺时针顺序依次将这 n 个机器人工厂编号为 $1 \sim n$ ，因为马路是环形的，所以第 n 个机器人工厂和第 1 个机器人工厂是由一段马路连接在一起的。小新将连接机器人工厂的这 n 段马路也编号为 $1 \sim n$ ，并规定第 i 段马路连接第 i 个机器人工厂和第 $i+1$ 个机器人工厂 ($1 \leq i \leq n-1$)，第 n 段马路连接第 n 个机器人工厂和第 1 个机器人工厂。

游戏过程中，每个单位时间内，每段马路上都会出现一些金币，金币的数量会随时间发生变化，即不同单位时间内同一段马路上出现的金币数量可能是不同的。小新需要机器人的帮助才能收集到马路上的金币。所需的机器人必须在机器人工厂用一些金币来购买，机器人一旦被购买，便会沿着环形马路按顺时针方向一直行走，在每个单位时间内行走一次，即从当前所在的机器人工厂到达相邻的下一个机器人工厂，并将经过的马路上的所有金币收集给小新，例如，小新在 i ($1 \leq i \leq n$) 号机器人工厂购买了一个机器人，这个机器人会从 i 号机器人工厂开始，顺时针在马路上

题目描述

行走，第一次行走会经过 i 号马路，到达 $i+1$ 号机器人工厂（如果 $i=n$ ，机器人会到达第 1 个机器人工厂），并将 i 号马路上的所有金币收集给小新。游戏中，环形马路上不能同时存在 2 个或者 2 个以上的机器人，并且每个机器人最多能够在环形马路上行走 p 次。小新购买机器人的同时，需要给这个机器人设定行走次数，行走次数可以为 $1 \sim p$ 之间的任意整数。当马路上的机器人行走完规定的次数之后会自动消失，小新必须立刻在任意一个机器人工厂中购买一个新的机器人，并给新的机器人设定新的行走次数。以下是游戏的一些补充说明：

- ① 游戏从小新第一次购买机器人开始计时。
- ② 购买机器人和设定机器人的行走次数是瞬间完成的，不需要花费时间。
- ③ 购买机器人和机器人行走是两个独立的过程，机器人行走时不能购买机器人，购买完机器人并且设定机器人行走次数之后机器人才能行走。
- ④ 在同一个机器人工厂购买机器人的花费是相同的，但是在不同机器人工厂购买机器人的花费不一定相同。

购买机器人花费的金币，在游戏结束时再从小新收集的金币中扣除，所以在游戏过

题目描述

程中小新不用担心因金币不足，无法购买机器人而导致游戏无法进行。也因为如此，游戏结束后，收集的金币数量可能为负。现在已知每段马路上每个单位时间内出现的金币数量和在每个机器人工厂购买机器人需要的花费，请你告诉小新，经过 m 个单位时间后，扣除购买机器人的花费，小新最多能收集到多少金币。

输入格式

第一行 3 个正整数， $n\ m\ p$ ，意义如题目所述。

接下来的 n 行，每行有 m 个正整数，每两个整数之间用一个空格隔开，其中第 i 行描述了 i 号马路上每个单位时间内出现的金币数量 ($1 \leq \text{金币数量} \leq 100$)，即第 i 行的第 j ($1 \leq j \leq m$) 个数表示第 j 个单位时间内 i 号马路上出现的金币数量。

最后一行，有 n 个整数，每两个整数之间用一个空格隔开，其中第 i 个数表示在 i 号机器人工厂购买机器人需要花费的金币数量 ($1 \leq \text{金币数量} \leq 100$)。

输出格式

1 个整数表示在 m 个单位时间内扣除购买机器人花费金币最多能收集到的金币数。

吐槽

吐槽

这道题目这么长，并且难得要命？

吐槽

这道题目这么长，并且难得要命？dalao 上来讲讲思路？

这道题目这么长，并且难得要命？dalao 上来讲讲思路？

pre 题解

吐槽

这道题目这么长，并且难得要命？dalao 上来讲讲思路？

pre 题解

我们很容易设计出状态 $f[i]$ 表示时间 i 可以获得的最大金钱。

吐槽

这道题目这么长，并且难得要命？dalao 上来讲讲思路？

pre 题解

我们很容易设计出状态 $f[i]$ 表示时间 i 可以获得的最大金钱。DP 的一贯套路。

吐槽

这道题目这么长，并且难得要命？dalao 上来讲讲思路？

pre 题解

我们很容易设计出状态 $f[i]$ 表示时间 i 可以获得的最大金钱。DP 的一贯套路。然后呢？我们考虑需要枚举什么。

吐槽

这道题目这么长，并且难得要命？dalao 上来讲讲思路？

pre 题解

我们很容易设计出状态 $f[i]$ 表示时间 i 可以获得的最大金钱。DP 的一贯套路。然后呢？我们考虑需要枚举什么。时间 i ，到达的城市 j 和这一步走的步数 k 。

吐槽

这道题目这么长，并且难得要命？dalao 上来讲讲思路？

pre 题解

我们很容易设计出状态 $f[i]$ 表示时间 i 可以获得的最大金钱。DP 的一贯套路。然后呢？我们考虑需要枚举什么。时间 i ，到达的城市 j 和这一步走的步数 k 。题目中给出来了，为什么不用？

吐槽

这道题目这么长，并且难得要命？dalao 上来讲讲思路？

pre 题解

我们很容易设计出状态 $f[i]$ 表示时间 i 可以获得的最大金钱。DP 的一贯套路。然后呢？我们考虑需要枚举什么。时间 i ，到达的城市 j 和这一步走的步数 k 。题目中给出来了，为什么不用？推一下状态转移方程？

吐槽

这道题目这么长，并且难得要命？dalao 上来讲讲思路？

pre 题解

我们很容易设计出状态 $f[i]$ 表示时间 i 可以获得的最大金钱。DP 的一贯套路。然后呢？我们考虑需要枚举什么。时间 i ，到达的城市 j 和这一步走的步数 k 。题目中给出来了，为什么不用？

推一下状态转移方程？

我们令 $cost[i]$ 表示第 i 个工厂机器人的雇佣费用， $r[j][i]$ 表示第 i 个时刻到达第 j 个工厂的价值， $sum[i][j]$ 表示价值的前缀和。

吐槽

这道题目这么长，并且难得要命？dalao 上来讲讲思路？

pre 题解

我们很容易设计出状态 $f[i]$ 表示时间 i 可以获得的最大金钱。DP 的一贯套路。然后呢？我们考虑需要枚举什么。时间 i ，到达的城市 j 和这一步走的步数 k 。题目中给出来了，为什么不用？

推一下状态转移方程？

我们令 $cost[i]$ 表示第 i 个工厂机器人的雇佣费用， $r[j][i]$ 表示第 i 个时刻到达第 j 个工厂的价值， $sum[i][j]$ 表示价值的前缀和。

考虑到一个事实，我们要在一个环上进行 DP，如何才能方便地转移？

吐槽

这道题目这么长，并且难得要命？dalao 上来讲讲思路？

pre 题解

我们很容易设计出状态 $f[i]$ 表示时间 i 可以获得的最大金钱。DP 的一贯套路。然后呢？我们考虑需要枚举什么。时间 i ，到达的城市 j 和这一步走的步数 k 。题目中给出来了，为什么不用？

推一下状态转移方程？

我们令 $cost[i]$ 表示第 i 个工厂机器人的雇佣费用， $r[j][i]$ 表示第 i 个时刻到达第 j 个工厂的价值， $sum[i][j]$ 表示价值的前缀和。

考虑到一个事实，我们要在一个环上进行 DP，如何才能方便地转移？

显然可以用将第一个城市编号为 0，然后将 i 对 n 取模的方式做到将第 n 转移到第 1 个城市。

吐槽

这道题目这么长，并且难得要命？dalao 上来讲讲思路？

pre 题解

我们很容易设计出状态 $f[i]$ 表示时间 i 可以获得的最大金钱。DP 的一贯套路。然后呢？我们考虑需要枚举什么。时间 i ，到达的城市 j 和这一步走的步数 k 。题目中给出来了，为什么不用？

推一下状态转移方程？

我们令 $cost[i]$ 表示第 i 个工厂机器人的雇佣费用， $r[j][i]$ 表示第 i 个时刻到达第 j 个工厂的价值， $sum[i][j]$ 表示价值的前缀和。

考虑到一个事实，我们要在一个环上进行 DP，如何才能方便地转移？

显然可以用将第一个城市编号为 0，然后将 i 对 n 取模的方式做到将第 n 转移到第 1 个城市。

对于价值，由于存在两个时间和金钱两个约束，我们需要进行一次二维前缀和。

吐槽

这道题目这么长，并且难得要命？dalao 上来讲讲思路？

pre 题解

我们很容易设计出状态 $f[i]$ 表示时间 i 可以获得的最大金钱。DP 的一贯套路。然后呢？我们考虑需要枚举什么。时间 i ，到达的城市 j 和这一步走的步数 k 。题目中给出来了，为什么不用？

推一下状态转移方程？

我们令 $cost[i]$ 表示第 i 个工厂机器人的雇佣费用， $r[j][i]$ 表示第 i 个时刻到达第 j 个工厂的价值， $sum[i][j]$ 表示价值的前缀和。

考虑到一个事实，我们要在一个环上进行 DP，如何才能方便地转移？

显然可以用将第一个城市编号为 0，然后将 i 对 n 取模的方式做到将第 n 转移到第 1 个城市。

对于价值，由于存在两个时间和金钱两个约束，我们需要进行一次二维前缀和。就是说， sum 值沿对角线传递。

$O(n^3)$ 题解

$O(n^3)$ 题解

预处理完这些数组以后，我们可以得到以下状态转移方程：

$$f[i] = \max\{f[i - k] + \text{sum}[i][j] - \text{sum}[i - k][j - k] - \text{cost}[j - k]\}$$

$O(n^3)$ 题解

预处理完这些数组以后，我们可以得到以下状态转移方程：

$$f[i] = \max\{f[i - k] + \text{sum}[i][j] - \text{sum}[i - k][j - k] - \text{cost}[j - k]\}$$

需要解释一下吗？

$O(n^3)$ 题解

预处理完这些数组以后，我们可以得到以下状态转移方程：

$$f[i] = \max\{f[i - k] + \text{sum}[i][j] - \text{sum}[i - k][j - k] - \text{cost}[j - k]\}$$

需要解释一下吗？

因为我们要求出在时刻 i 走到第 j 个工厂的最大价值，我们就要知道第 $j - k$ 个工厂的最大价值，即 $f[j - k]$ ；然后加上这 $j - k$ 步的经过工厂的价值，由于是预处理的前缀和数组，我们可以得到 $j - k$ 步经过工厂的价值实际上是时间区间 $[0, i]$ 与时间区间 $[0, i - k]$ 从第 $j - k$ 个工厂走到第 i 个工厂价值的差，再减掉雇佣第 $j - k$ 号工厂机器人的费用。

$O(n^3)$ 题解

预处理完这些数组以后，我们可以得到以下状态转移方程：

$$f[i] = \max\{f[i - k] + \text{sum}[i][j] - \text{sum}[i - k][j - k] - \text{cost}[j - k]\}$$

需要解释一下吗？

因为我们要求出在时刻 i 走到第 j 个工厂的最大价值，我们就要知道第 $j - k$ 个工厂的最大价值，即 $f[j - k]$ ；然后加上这 $j - k$ 步的经过工厂的价值，由于是预处理的前缀和数组，我们可以得到 $j - k$ 步经过工厂的价值实际上是时间区间 $[0, i]$ 与时间区间 $[0, i - k]$ 从第 $j - k$ 个工厂走到第 i 个工厂价值的差，再减掉雇佣第 $j - k$ 号工厂机器人的费用。

然后依次枚举 i, j, k 就行了，可以拿到 90 分的好成绩。

$O(n^3)$ 题解

预处理完这些数组以后，我们可以得到以下状态转移方程：

$$f[i] = \max\{f[i - k] + \text{sum}[i][j] - \text{sum}[i - k][j - k] - \text{cost}[j - k]\}$$

需要解释一下吗？

因为我们要求出在时刻 i 走到第 j 个工厂的最大价值，我们就要知道第 $j - k$ 个工厂的最大价值，即 $f[j - k]$ ；然后加上这 $j - k$ 步的经过工厂的价值，由于是预处理的前缀和数组，我们可以得到 $j - k$ 步经过工厂的价值实际上是时间区间 $[0, i]$ 与时间区间 $[0, i - k]$ 从第 $j - k$ 个工厂走到第 i 个工厂价值的差，再减掉雇佣第 $j - k$ 号工厂机器人的费用。

然后依次枚举 i, j, k 就行了，可以拿到 90 分的好成绩。

是不是感觉自己很棒棒啊？

优化

优化

我们对刚才的 DP 方程稍加处理，把不需要枚举 k 的一项提取出来？

优化

我们对刚才的 DP 方程稍加处理，把不需要枚举 k 的一项提取出来？

$$f[i] = \max(f[i-k] - \text{sum}[i-k][j-k] - \text{cost}[j-k]) + \text{sum}[i][j]$$

优化

我们对刚才的 DP 方程稍加处理，把不需要枚举 k 的一项提取出来？

$$f[i] = \max(f[i-k] - \text{sum}[i-k][j-k] - \text{cost}[j-k]) + \text{sum}[i][j]$$

注意到在转移过程中 i 和 j 都用到了 k 这个不变量，我们令 $g[i][j] = f[i] - \text{sum}[i][j] - \text{cost}[j]$ ，状态转移方程可以转化为以下形式：

优化

我们对刚才的 DP 方程稍加处理，把不需要枚举 k 的一项提取出来？

$$f[i] = \max(f[i-k] - \text{sum}[i-k][j-k] - \text{cost}[j-k]) + \text{sum}[i][j]$$

注意到在转移过程中 i 和 j 都用到了 k 这个不变量，我们令 $g[i][j] = f[i] - \text{sum}[i][j] - \text{cost}[j]$ ，状态转移方程可以转化为以下形式：

$$f[i] = \max(g[i-k][j-k]) + \text{sum}[i][j]$$

优化

我们对刚才的 DP 方程稍加处理，把不需要枚举 k 的一项提取出来？

$$f[i] = \max(f[i-k] - \text{sum}[i-k][j-k] - \text{cost}[j-k]) + \text{sum}[i][j]$$

注意到在转移过程中 i 和 j 都用到了 k 这个不变量，我们令 $g[i][j] = f[i] - \text{sum}[i][j] - \text{cost}[j]$ ，状态转移方程可以转化为以下形式：

$$f[i] = \max(g[i-k][j-k]) + \text{sum}[i][j]$$

就是对 g 数组的一条对角线求个最大值，然后加上价值的前缀和。

优化

我们对刚才的 DP 方程稍加处理，把不需要枚举 k 的一项提取出来？

$$f[i] = \max(f[i-k] - \text{sum}[i-k][j-k] - \text{cost}[j-k]) + \text{sum}[i][j]$$

注意到在转移过程中 i 和 j 都用到了 k 这个不变量，我们令 $g[i][j] = f[i] - \text{sum}[i][j] - \text{cost}[j]$ ，状态转移方程可以转化为以下形式：

$$f[i] = \max(g[i-k][j-k]) + \text{sum}[i][j]$$

就是对 g 数组的一条对角线求个最大值，然后加上价值的前缀和。

所以我们可以先预处理出 g 数组，直接把这个数组同一条对角线上的值扔到一个优先队列里去。然后只需要枚举 i, j 就能 $O(1)$ 转移了。

优化

我们对刚才的 DP 方程稍加处理，把不需要枚举 k 的一项提取出来？

$$f[i] = \max(f[i-k] - \text{sum}[i-k][j-k] - \text{cost}[j-k]) + \text{sum}[i][j]$$

注意到在转移过程中 i 和 j 都用到了 k 这个不变量，我们令 $g[i][j] = f[i] - \text{sum}[i][j] - \text{cost}[j]$ ，状态转移方程可以转化为以下形式：

$$f[i] = \max(g[i-k][j-k]) + \text{sum}[i][j]$$

就是对 g 数组的一条对角线求个最大值，然后加上价值的前缀和。

所以我们可以先预处理出 g 数组，直接把这个数组同一条对角线上的值扔到一个优先队列里去。然后只需要枚举 i, j 就能 $O(1)$ 转移了。

总的时间复杂度 $O(n^2 \log n)$ ，可以通过全部的数据。

优化

我们对刚才的 DP 方程稍加处理，把不需要枚举 k 的一项提取出来？

$$f[i] = \max(f[i-k] - \text{sum}[i-k][j-k] - \text{cost}[j-k]) + \text{sum}[i][j]$$

注意到在转移过程中 i 和 j 都用到了 k 这个不变量，我们令 $g[i][j] = f[i] - \text{sum}[i][j] - \text{cost}[j]$ ，状态转移方程可以转化为以下形式：

$$f[i] = \max(g[i-k][j-k]) + \text{sum}[i][j]$$

就是对 g 数组的一条对角线求个最大值，然后加上价值的前缀和。

所以我们可以先预处理出 g 数组，直接把这个数组同一条对角线上的值扔到一个优先队列里去。然后只需要枚举 i, j 就能 $O(1)$ 转移了。

总的时间复杂度 $O(n^2 \log n)$ ，可以通过全部的数据。

也可以用单调队列优化。一样的道理，注意到所求区间为长度为 k 的滑动区间。

优化

我们对刚才的 DP 方程稍加处理，把不需要枚举 k 的一项提取出来？

$$f[i] = \max(f[i-k] - \text{sum}[i-k][j-k] - \text{cost}[j-k]) + \text{sum}[i][j]$$

注意到在转移过程中 i 和 j 都用到了 k 这个不变量，我们令 $g[i][j] = f[i] - \text{sum}[i][j] - \text{cost}[j]$ ，状态转移方程可以转化为以下形式：

$$f[i] = \max(g[i-k][j-k]) + \text{sum}[i][j]$$

就是对 g 数组的一条对角线求个最大值，然后加上价值的前缀和。

所以我们可以先预处理出 g 数组，直接把这个数组同一条对角线上的值扔到一个优先队列里去。然后只需要枚举 i, j 就能 $O(1)$ 转移了。

总的时间复杂度 $O(n^2 \log n)$ ，可以通过全部的数据。

也可以用单调队列优化。一样的道理，注意到所求区间为长度为 k 的滑动区间。滑动窗口问题？

优化

我们对刚才的 DP 方程稍加处理，把不需要枚举 k 的一项提取出来？

$$f[i] = \max(f[i-k] - \text{sum}[i-k][j-k] - \text{cost}[j-k]) + \text{sum}[i][j]$$

注意到在转移过程中 i 和 j 都用到了 k 这个不变量，我们令 $g[i][j] = f[i] - \text{sum}[i][j] - \text{cost}[j]$ ，状态转移方程可以转化为以下形式：

$$f[i] = \max(g[i-k][j-k]) + \text{sum}[i][j]$$

就是对 g 数组的一条对角线求个最大值，然后加上价值的前缀和。

所以我们可以先预处理出 g 数组，直接把这个数组同一条对角线上的值扔到一个优先队列里去。然后只需要枚举 i, j 就能 $O(1)$ 转移了。

总的时间复杂度 $O(n^2 \log n)$ ，可以通过全部的数据。

也可以用单调队列优化。一样的道理，注意到所求区间为长度为 k 的滑动区间。滑动窗口问题？

所以预处理完了之后扔到 n 个单调队列里，时间复杂度可以降低为 $O(n^2)$ 。

谢谢

谢谢

提问时间

有什么问题吗？

谢谢

提问时间

有什么问题吗？

作业：完成今天讲的十道例题。