Two Dishes DAY2

SORTREEW

問題概要

問題文を読みましょう

小課題1

制約：制限時間(SやT)の制約が全て同じ

取る順序を気にしなくてよくなる。つまり、カレーと丼を制限時間内に取った個数の組から答えが求まる。また、カレーを取った個数を固定すると丼を取った個数は一意に定まり、これは二分探索で求められる。丼を固定する場合もやる。O(NlogM+MlogN)

あまりに多い小課題

	N,M<=12	N,M<=2000	N,M<=200000	N,M<=1000000
P,Q=1	2	3	4	
P,Qは正			5	6
P,Qの制約なし			7	満点！

考察

空白の時間はなく、丼やカレーの中では手順が決まっているので、

ある手順で点数が取れるかどうかは

もう一種類の食べ物の手順がどこまで進んでいるかだけで決まる（これまでのとり方の順序は関係なく）

問題の変換

N＼M	0	1	2	3
0	0	1	2	3
1	0	1	2	3
2	1	2	3	4
3	2	3	4	5
4	3	4	5	6

N*Mのグリッドがあり、
マス目の間に左や上から直線が伸びている。

i-1行目とi行目の間の線を通るとP_i点、
i-1列目とi列目の間の線を通るとQ_i点、もらうことができる
左上から右下に行く時、得られる得点の最大値を求めよ。

各行と列ごとに、もう一方の食べ物を何回食べてもいいか（列や行を何回通ってもいいか）を二分探索や累積和を用いて計算することで、この表を作成することができる。

小課題2

制約：N<=12,M<=12

とり方を全探索する

O(2^(N+M))

小課題3

制約：N<=2000,M<=2000

とり方を動的計画法で求める。

よくあるグリッド上のDP

小課題3

制約：N<=2000,M<=2000

更新式は

DP[n][m]=max(DP[n-1][m]+f1(n,m),DP[n][m-1]+f2(n,m))

f1(n,m)=上からn,mに来た時、線を通るならP[n]、そうでないなら0

f2(n,m)=左からn,mに来た時、線を通るならQ[m]、そうでないなら0

O(NM)

ここまでは春合宿に来れる人なら取れるはず（取ってほしい）

小課題4,5,6

主な制約：**P,Qは正**（これがうれしい制約です）

とり方を動的計画法で求めたいが、制約がやばい。しかし動的計画法以外で解ける気もしないので、テーブルを見てみる

小課題4,5,6

主な制約：P,Qは正（これがうれしい制約です）

特徴：DP[n][m]-DP[n][m-1]は行ごとに見るとあまり変化しない？（サンプル1）
このテーブルでも復元はできる
(オレンジ色の線を通りながら、横線の得点とテーブルの数字を足し合わせる)

N＼M	0	1	2	3
0	0	1	2	3
1	0	1	2	3
2	1	2	3	4
3	2	3	4	5
4	3	4	5	6

N＼M	0	1	2	3
0		1	1	1
1		1	1	1
2		1	1	1
3		1	1	1
4		1	1	1

小課題4,5,6

テーブルの変化が少ないとうれしい

何がうれしいか？

→DP[n][]にDP[n-1][]のテーブルを使いまわして、更新をサボることで高速化できることがある。

ここからテーブルの使い回しをすることにして、

より変化が少なく、元のDPテーブルに復元できるような新たなDPテーブルを考えていく

他のサンプルでも試す

特徴：DP[n][m]-DP[n][m-1]はあまり変化しない(右図）

N＼M	0	1 (+2)	2 (+7)	3 (+19)	4 (+4)	5 (+15)	6 (+14)	7 (+8)
0	0	2	9	28	32	47	47	47
1(+16)	16	18	25	44	48	63	63	63
2(+10)	26	28	35	54	54	63	63	63
3(+1)	27	29	36	54	54	63	63	63
4(+16)	43	45	45	54	54	63	63	63
5(+10)	43	45	45	54	54	63	63	63

N＼M	0	1 (+2)	2 (+7)	3 (+19)	4 (+4)	5 (+15)	6 (+14)	7 (+8)
0	0	2	7	19	4	15	0	0
1(+16)	0	2	7	19	4	15	0	0
2(+10)	0	2	7	19	0	9	0	0
3(+1)	0	2	7	18	0	9	0	0
4(+16)	0	2	0	9	0	9	0	0
5(+10)	0	2	0	9	0	9	0	0

Q:こんなの気づけなくない？

A:はい

ただ、

DP[n][m]=max(DP[n-1][m]+f1(n,m),DP[n][m-1]+f2(n,m))

からDP[n][m]はmax内の2つの式いずれかと等しく、f1やf2はn,mそれぞれ固定すれば2種類しか値を取らないので、差分を取ると変化少ない気持ちにはなる（かもしれない）

他のサンプルでも試す

特徴2:

マスの左に線があるなら

DP[n][m]-DP[n][m-1]=Q[m]

偶然か？

N＼M	0	1 (+2)	2 (+7)	3 (+19)	4 (+4)	5 (+15)	6 (+14)	7 (+8)
0	0	2	7	19	4	15	0	0
1(+16)	0	2	7	19	4	15	0	0
2(+10)	0	2	7	19	0	9	0	0
3(+1)	0	2	7	18	0	9	0	0
4(+16)	0	2	0	9	0	9	0	0
5(+10)	0	2	0	9	0	9	0	0

他のサンプルでも試す

特徴2:

マスの左に線があるなら

DP[n][m]-DP[n][m-1]=Q[m]

理由:横線はできるだけ左にいたほうが通りやすい

（短期的には点がもらえなくなるタイミングまで取るのを先送りにしたほうがよい）

N＼M	0	1 (+2)	2 (+7)	3 (+19)	4 (+4)	5 (+15)	6 (+14)	7 (+8)
0	0	2	7	19	4	15	0	0
1(+16)	0	2	7	19	4	15	0	0
2(+10)	0	2	7	19	0	9	0	0
3(+1)	0	2	7	18	0	9	0	0
4(+16)	0	2	0	9	0	9	0	0
5(+10)	0	2	0	9	0	9	0	0

テーブルの変形

つまり、DP[n][m]-DP[n][m-1]-Q[m]も変化が少なそう？

N＼M	0	1 (+2)	2 (+7)	3 (+19)	4 (+4)	5 (+15)	6 (+14)	7 (+8)
0	0	2	7	19	4	15	0	0
1(+16)	0	2	7	19	4	15	0	0
2(+10)	0	2	7	19	0	9	0	0
3(+1)	0	2	7	18	0	9	0	0
4(+16)	0	2	0	9	0	9	0	0
5(+10)	0	2	0	9	0	9	0	0

N＼M	0	1 (+2)	2 (+7)	3 (+19)	4 (+4)	5 (+15)	6 (+14)	7 (+8)
0	0	0	0	0	0	0	0	0
1(+16)	0	0	0	0	0	0	0	0
2(+10)	0	0	0	0	-4	-6	0	0
3(+1)	0	0	0	-1	-4	-6	0	0
4(+16)	0	0	-7	-10	-4	-6	0	0
5(+10)	0	0	-7	-10	-4	-6	0	0

テーブルの変形

変化は少ないが、

0以上というよさそうな性質が崩れるので、

左に線がないマスには DP[n][m]-DP[n][m-1]を入れてみる(これでも復元できる)

N＼M	0	1 (+2)	2 (+7)	3 (+19)	4 (+4)	5 (+15)	6 (+14)	7 (+8)
0	0	0	0	0	0	0	0	0
1(+16)	0	0	0	0	0	0	0	0
2(+10)	0	0	0	0	0	9	0	0
3(+1)	0	0	0	18	0	9	0	0
4(+16)	0	0	0	9	0	9	0	0
5(+10)	0	2	0	9	0	9	0	0

テーブルの変形

変化は少ないが、

0以上というよさそうな性質が崩れるので、

左に線がないマスには DP[n][m]-DP[n][m-1]を入れてみる(これでも復元できる)

N＼M	0	1 (+2)	2 (+7)	3 (+19)	4 (+4)	5 (+15)	6 (+14)	7 (+8)
0	0	0	0	0	0	0	0	0
1(+16)	0	0	0	0	0	0	0	0
2(+10)	0	0	0	0	0	9	0	0
3(+1)	0	0	0	18	0	9	0	0
4(+16)	0	0	0	9	0	9	0	0
5(+10)	0	2	0	9	0	9	0	0

テーブルの変形

変化は少ないが、

0以上というよさそうな性質が崩れるので、

左に線がないマスには DP[n][m]-DP[n][m-1]を入れてみる(これでも復元できる)

N＼M	0	1 (+2)	2 (+7)	3 (+19)	4 (+4)	5 (+15)	6 (+14)	7 (+8)
0	0	0	0	0	0	0	0	0
1(+16)	0	0	0	0	0	0	0	0
2(+10)	0	0	0	0	0	9	0	0
3(+1)	0	0	0	18	0	9	0	0
4(+16)	0	0	0	9	0	9	0	0
5(+10)	0	2	0	9	0	9	0	0

テーブルの更新

さて、どうやって更新するか（これが難しい）

テーブルの更新

縦の更新と横の更新を独立に見たいので、極端な例を試してみる

N＼M	0	1 (+2)	2 (+4)	3 (+6)	4 (+8)	5 (+10)	6 (+12)
0	0	2	6	12	20	30	42
1(+1)	1	2	6	12	20	30	42
2(+3)	4	4	6	12	20	30	42
3(+5)	9	9	9	12	20	30	42
4(+7)	16	16	16	16	20	30	42
5(+9)	25	25	25	25	25	30	42

N＼M	0	1 (+2)	2 (+4)	3 (+6)	4 (+8)	5 (+10)	6 (+12)
0		0	0	0	0	0	0
1(+1)		1	4	6	8	10	12
2(+3)		0	2	6	8	10	12
3(+5)		0	0	3	8	10	12
4(+7)		0	0	0	4	10	12
5(+9)		0	0	0	0	5	12

極端な例

（おさらい）テーブルのn行m列の式Diff[n][m]は

左に線がある:

Diff[n][m]=

DP[n][m]-DP[n][m]-Q[m]=0

左に線がない:

Diff[n][m]=

DP[n][m]-DP[n][m]

N＼M	0	1 (+2)	2 (+4)	3 (+6)	4 (+8)	5 (+10)	6 (+12)
0		0	0	0	0	0	0
1(+1)		1	4	6	8	10	12
2(+3)		0	2	6	8	10	12
3(+5)		0	0	3	8	10	12
4(+7)		0	0	0	4	10	12
5(+9)		0	0	0	0	5	12

わかったこと

縦の線が消えた時：

消えた線の右下のdiffに、

その線の点(Q[m])が加算される

理由：差は埋まらず、単純にテーブルの式が変わるため

N＼M	0	1 (+2)	2 (+4)	3 (+6)	4 (+8)	5 (+10)	6 (+12)
0		0	0	0	0	0	0
1(+1)		1	4	6	8	10	12
2(+3)		0	2	6	8	10	12
3(+5)		0	0	3	8	10	12
4(+7)		0	0	0	4	10	12
5(+9)		0	0	0	0	5	12

わかったこと

横の線が消えた時：

消えた線の右下を(n,m)とする

DP[n][m]-DP[n][m-1]>=P[n]のとき、

DP[n][m]は変わらないが、

DP[n][m-1]はk増えるので

Diff[n][m]はk減少する。

N＼M	0	1 (+2)	2 (+4)	3 (+6)	4 (+8)	5 (+10)	6 (+12)
0		0	0	0	0	0	0
1(+1)		1	4	6	8	10	12
2(+3)		0	2	6	8	10	12
3(+5)		0	0	3	8	10	12
4(+7)		0	0	0	4	10	12
5(+9)		0	0	0	0	5	12

わかったこと

横の線が消えた時

（おおざっぱに）:

DP[n][m]-DP[n][m-1]<kのとき、

DP[n][m]は横から遷移するようになる。

さらに、DP[n][m]の値も余った分だけ増えて、右のほうに伝搬していく

N＼M	0	1 (+2)	2 (+4)	3 (+6)	4 (+8)	5 (+10)	6 (+12)
0		0	0	0	0	0	0
1(+1)		1	4	6	8	10	12
2(+3)		0	2	6	8	10	12
3(+5)		0	0	3	8	10	12
4(+7)		0	0	0	4	10	12
5(+9)		0	0	0	0	5	12

わかったこと

横の線が消えた時

（くわしく）:

消えた線の右下を(n,m)とする

DP[n][m]-DP[n][m-1]<P[n]のとき、

DP[n][m] =DP[n][m-1]+f(n,m)となる、

DP[n][m-1]はk増え、DP[n][m]もQ[m]-diff[n][m]増える

Diff[n][m]は0になる。

N＼M	0	1 (+2)	2 (+4)	3 (+6)	4 (+8)	5 (+10)	6 (+12)
0		0	0	0	0	0	0
1(+1)		1	4	6	8	10	12
2(+3)		0	2	6	8	10	12
3(+5)		0	0	3	8	10	12
4(+7)		0	0	0	4	10	12
5(+9)		0	0	0	0	5	12

わかったこと

横の線が消えた時

（くわしく）:

さらに左から遷移しているDPの値が全て増えるので、次に上から遷移しているマスで今までの議論をQ[m]の代わりにQ[m]-diff[n][m]を用いて同様に行う。

N＼M	0	1 (+2)	2 (+4)	3 (+6)	4 (+8)	5 (+10)	6 (+12)
0		0	0	0	0	0	0
1(+1)		1	4	6	8	10	12
2(+3)		0	2	6	8	10	12
3(+5)		0	0	3	8	10	12
4(+7)		0	0	0	4	10	12
5(+9)		0	0	0	0	5	12

わかったこと

計算量は？

まず、値の加算は縦の線の数ぶんだけ、つまりO(M)回

値の減算は伝搬があるが、減算する場所がO(M)箇所しかなく、1度の減算でN回減算するとN-1箇所消えるのでならすと

値の減少はO(N+M)回

N＼M	0	1 (+2)	2 (+4)	3 (+6)	4 (+8)	5 (+10)	6 (+12)
0		0	0	0	0	0	0
1(+1)		1	4	6	8	10	12
2(+3)		0	2	6	8	10	12
3(+5)		0	0	3	8	10	12
4(+7)		0	0	0	4	10	12
5(+9)		0	0	0	0	5	12

わかったこと

計算量は？

減算する場所の管理はsetなどを使うと全体でO(MlogM)くらいでできる

全体でO((N+M)log(N+M))とかで抑えられる

N＼M	0	1 (+2)	2 (+4)	3 (+6)	4 (+8)	5 (+10)	6 (+12)
0		0	0	0	0	0	0
1(+1)		1	4	6	8	10	12
2(+3)		0	2	6	8	10	12
3(+5)		0	0	3	8	10	12
4(+7)		0	0	0	4	10	12
5(+9)		0	0	0	0	5	12

小課題4,5

伝搬を1回分しか処理できない（実装や考察不足など）
→P,Q=1の小課題だけ正解
diffを取らず、セグ木の区間加算等logを余分につける
→N,M<=200000だけ正解

小課題7 満点解法

P,Qに制約がない場合を考える

線が消えるときの、diffに対する操作がちゃんとできればいい

今の理解では操作が怪しくなるので、操作について考え直す

わかったこと(みなおし)

縦の線が消えた時
(Q[m]>0):

消えた線の右下のdiffに、

その線の点(Q[m])が加算される

理由:上からの遷移はどうしても縦の線を通るが、左からの遷移は縦の線を通らないため

N＼M	0	1 (+2)	2 (+4)	3 (+6)	4 (+8)	5 (+10)	6 (+12)
0		0	0	0	0	0	0
1(+1)		1	4	6	8	10	12
2(+3)		0	2	6	8	10	12
3(+5)		0	0	3	8	10	12
4(+7)		0	0	0	4	10	12
5(+9)		0	0	0	0	5	12

満点解法

diffに対する操作対する影響は(得点が正なら）同じ

得点が負になる場合を考える

よく考えると、今まで減っていた部分がかわりに増えて、増えていた部分が減ることになることがわかる。

満点解法

操作を考える（行）

i行目のP[i]>0の横線が消える：

線の右下にあるマスに減算（伝搬する）

i行目のP[i]<0の横線が消える：

線の右下にあるマスに加算

満点解法

操作を考える（列）

j列目のQ[j]>0の縦線が消える：

線の右下にあるマスに加算

j列目のQ[j]<0の縦線が消える：

線の右下にあるマスに減算（伝搬する）

満点解法

非対称に見えた操作の原因は行か列かの違いではなく、正か負かの違いにあった。

（これはdiffの下限が0であることに起因している？）

満点解法

以下のクエリを投げて処理していく

Add(x,v):diff[x]にvを加算

Decrease(x,v):

diff[x]<vのとき:diff[x]=0としてy>xかつdiff[y]となる最小のyに対し、Decrease(y,v-diff[x])をする

diff[x]>=vのとき:diff[x]にvを減算

満点解法

操作を抽象的に考える（行）

i行目のP[i]>0の横線が消える：

Decrease(x,P[i])

i行目のP[i]<0の横線が消える：

Add(x,-P[i])

xは線の右下の座標

満点解法

操作を抽象的に考える（列）

j列目のQ[j]>0の縦線が消える：

Add(x,Q[j])

j列目のQ[j]<0の縦線が消える：

Decrease(x,-Q[j])

xは線の右下の座標

満点解法

これで全ての更新を処理することができ、線の情報からDP[n][m]-DP[n][m-1]のテーブルに復元して、更に左下を経由するルートを使ってDP[N][M]を復元できる。

計算量は加算と減算がそれぞれならしでO(N+M)回でできるので、O((N+M)log(N+M))で抑えられる

実装上の注意

マスに対する操作と線に対する操作があるので、気をつけないと添字に+1が大量発生し、困る

行ごとに加算と減算をそれぞれ分けてやること（加算を先にやらないとWA）

得点分布

15点 11人
10点 5人
0点 2人
3点 1人
54点 1人
65点 1人