هوالحق

www.Ginoo.Ir

بازی زیر را در نظر بگیرید. تعدادی کارت وجود دارد که روی هر کدام یک عدد نوشته شده است. یک معامله گر یک توالی از کارت های $s\_{1},s\_{2}, ….,s\_{n}$ که روی هر کارت $s\_{i}$ عدد $v\_{i}$ نوشته شده است. سپس دو بازیگر هر کدام یک کارت از توالی را بر می دارند اما فقط میتوانند اولین یا آخرین کارت از باقیمانده کارت ها را بردارند. هدف انتخاب کارت هایی است که بزرگترین مجموع اعداد روی کارت را داشته باشند. فرض کنید که n زوج است. الگوریتم بهینه ای از مرتبه $O\left(n^{2}\right)$ برای بازیگر شماره یک ارائه کنید. با داشتن توالی اولیه, الگوریتم شما باید اطلاعاتی را از قبل با مرتبه درجه دو محاسبه کند و سپس بازیگر شماره یک باید بتواند بطور بهینه انتخاب ها را براساس اطلاعات محاسبه شده در زمان خطی انجام دهد.

برای حل این مسیله :

من در نظر گرفتم که اگه نفر اول کارت a رو انتخاب کنه و نفر دوم b رو انتخاب کنه پس باید a-b ماکسیمم بشه یعنی اختلاف انتخابها بیشترین بشه

برای حل از راه پویا رفتم و بصورت از بالا به پایین نوشتم به اینصورت که :

هردفعه یا نفر اول اولی رو برمیداره یا آخری رو حالا ما میخوایم بدونیم کدوم کارت رو برداره بهتره پس زیر مسیله ها رو اینجوری حل میکنیم که هردفعه اختلاف کارت نفر اول بیشتر از نفر دوم بشه

من فرض کردم نفر دوم هر دفعه دوست داره زیاده رو برداره پس ما نگاه میکنیم ببینیم کدوم کارت رو برداره بهتره من برا حلش :

struct m

{

 int first;

 int second;

}

;

m path[50][50];

m h;

یک استراکت گرفتم برای اینکه انتخابارو توش نگه دارم :

چون اولین ماله نفر اول میشه دومین ماله نفر دوم این استراکت هم دو مقداره است بعد یک آرایه ازش تعریف کردم که مسیر ها انتخاب رو نگه دارم.

int s = kart[i] - play(i + 1, j);

 int k = kart[j] - play(i, j - 1);

 record[i][j] = maxe(s, k);

 if (s > k)

 {

 path[i][j].first=i+1;

 path[i][j].second=j;

 }

 else

 {

 path[i][j].first=i;

 path[i][j].second=j-1;

 }

 return record[i][j];

همونطور که گفتم از اختلاف استفاده کردم I به اول کارت j به انتهای کارت اشاره داره اگه انتخابمون اولی باشه پس باید از I یکی کم شه j بمونه تا ببینیم انتخاب بعدی چطور بهتره واسه اینکه آخری هم انتخاب بشه از همین راه رفتم بعدش هرکدوم اختلاف ماکس داشت بهترینه پس نگهش داشتم به عنوان مسیره رفته بعدش record توش اختلافات رو نگه میداره بعدش آخرین خونش میشه اختلاف کارت بازیگر a و b پس a-b=diff

تو ورودی هم اومدم همه کارتا رو جمع زدم که میشه a+b=sum

حالا برای اینکه مجموع کارتهای نفر اول بدست بیاد از این دو رابطه استفاده کردم diff+sum)/2

رو نوشتم بعدش مسیر رو نگه داشتم :

برای اینکار وقتی یک کارت تک موند خود همون کارت میشه i==j

وقتی که هم I بزرگتر j شد یعنی کارتا تمومن

 پس باید حالا جدول مقدار گذاری بشن و تموم میشه

تو هر خونه آرایه چون از نوع استراکته دو مقدار ذخیره میشه

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | First/ second |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

برای چاپ مقادیر هم از یک رابطه بازگشتی استفاده کردم چون یک در میون نوبت نفر اول میشه t رو یکی هردفعه خونه ای که هستیم بصورت بازگشتی انتخاب بعدی اندیسش رو برای پیدا کردن تو آرایه بهمون میده در میون true کردم تا مقادیر رو چاپ کنه .

print(int t, int i, int j){

 h= path[i][j];

 if (t){

 if (h.first == i - 1)//inja in moghayese vase in anjam mishe ke bdune kodum kart entkhab shode I (kart aval )//j kart (akhar)

 cout<< kart[i];

 else

 cout<< kart[j];

 }

 if (i < j)

 print(!t, h.first, h.second);//

خط آخر هم واسه اینه که مقادیر بعدی رو تعیین کنه.