

(1.1)

رویکرد غیر هوشمندانه برای این سوال به این صورت است که در هر مرحله که نوبت ما میشود ما 2 انتخاب داریم. حال جند بار نوبت ما می شود؟ حدود $n/2$ پس ما باید $(n/2)^2$ حالت را بررسی کنیم تا حالت بهینه را پیدا کنیم. پس در حالت بروت فورس مساله ما از مرتبه زیر است: $O(2^n)$

(1.2)

راهکار هوشمندانه برای پاسخ به این مسئله به صورت حریصانه تعریف میشود. به این صورت زیر مسئله را تعریف میکنم:
4 خونه از ردیف را با ایندکس های 0 و 1 و $i-1$ و i در نظر میگیریم. ما بین 0 و i یکی را باید انتخاب کنیم. میایم در نظر میگیریم که اگر 0 رو انتخاب کنیم بعد حریف ماکسیمم 1 و i یکی رو انتخاب کرد اختلاف امتیاز بهتری با حریف میگیریم یا ما i رو انتخاب کنیم و حریف بین 0 و $i-1$ ماکسیمم رو انتخاب کنه. مثلا برای صورت سوال به این شکل میشه:

10 80 90 30

اگر ما 30 رو انتخاب کنیم رو انتخاب کنیم حریف بین 10 و 90 قاعدتا 90 رو انتخاب میکنه و اختلاف امتیاز ما با حریف 60 میشه. اگر ما 10 رو انتخاب کنیم حریف 80 رو انتخاب میکنه و اختلاف منفی 70 میشه. پس 30 رو انتخاب میکنیم.
حالا اگر بین 80 و 90 چیزی وجود داشت، ما در نظر نمیگیریم تا دور بعدی که رنج زیر مساله بهش برسه.
اگر 3 تا مهره مونده بود وسطی به حال ما فرقی نمیکرد پس بین چپی و راستی بیشترین رو بر میداریم و اگر 2 تا مونده بود صرفا فقط بیشترین رو بر میداریم. به این شکل زیر مساله رو برای الگوریتم حریصانه تعریف میکنیم و تا اخر مساله رو با این رویکرد جلو میبریم.

(1.3)

پیچیدگی زمانی بهینه برای این مساله از مرتبه $O(n)$ است

(2)

اگر مکان فعلی P باشد و شهر های x و y در مسیرمان باشند و بعد از y به Q بخواهیم برویم، اگر در شهر x سوخت گیری کنیم و به Q برسیم که فقط یکبار توقف کردیم و کمترین میزان ممکن توقف را داشته ایم. ولی اگر در y توقف کنیم باز هم به Q میرسیم ولی با سوخت بیشتر.
الگوریتم حریصانه را به این شکل تعریف میکنیم که ما اگر برای رفتن به شهر y به اندازه کافی سوخت داشتیم، به راه خود تا آن شهر ادامه میدهیم، اما اگر سوخت ما کمتر از میزان لازم برای رفتن از x به y بود در همان شهر x سوخت گیری میکنیم.

(3)

برای حل این مسئله میتوانیم از الگوریتم پریم با نقطه شروع هر یک از نود ها (شهر ها) استفاده کنیم و به این صورت به ترتیب از نزدیکترین شهر تا دورترین شهر را میتونیم به شهر مبدا سورت کنیم و به این شکل فاصله هر شهر تا شهر دیگر را بدست آوریم. روش دیگری که میتوان استفاده کرد و حتی با پریم ترکیب کرد داینامیک پروگرامینگ است. به این صورت که فاصله هر شهر شهر تا شهر های دیگر را که حساب میکنیم، اگر در روند حساب کردن یک شهر تا شهر دیگر اگر به فاصله 2 شهری که قبلا دیدیم برخوردیم، آن را بجای محاسبه از حافظه میخوانیم.
روش استفاده شده در کد ارسال شده بروت فورس است.
خروجی کد بصورت یک جدول $n \times n$ نمایش داده شده است.

(4)

این مسئله در واقع یک مسئله bin packagin problem است. به این صورت که این که 2 سارق همزمان میتوانند داخل سالن باشند در واقع 2 تا سبد داریم که هرکدام به اندازه مقدار بیرون بودن نگهبان جا دارد. با رویکرد best fit ما در 2 ظرف با ظرفیت G سعی میکنیم نیاز سارقین را جای دهیم. چرا best fit؟ چون باید تا جای ممکن هر دو ظرف پر شود.