

سوال جنگ سکه ها:

دو نفر هستند که هر کدام در هر نوبت میتونن یکی از دو سر آرایه را انتخاب کنند تا حداکثر سکه را جمع کنند. که در این مسئله شروع کننده هم محمدرضا است. این رو هم باید لحاظ کنیم که هر دو نفر بهینه بازی میکنند.

با استفاده از رویکرد دی پی با tabulation از یه جدول دو بعدی استفاده میکنیم و مقادیر را از زیرمسئله های کوچک به بزرگ پر میکنیم. $[Dp[n][n]$ میسازیم که $[dp[i][j]$ بیشترین سکه ای که بازیکن جاری میتونه از بازه ی i تا j بگیرد. حالت های ممکن را با فرمول ریاضی در جدول dp محاسبه میکنیم. که در نهایت بهترین تصمیم ممکن برای محمدرضا مشخص میشه. یه حلقه به نام gap میسازیم که نشون دهنده ی فاصله ی بین i, j است که از بازه های کوچیک شروع میکنیم تا به کل آرایه برسیم. گزینه هایی که حریف میتونه بعد از ما انتخاب کنه را لحاظ میکنیم. چون اون هم بهترین انتخاب را برای خودش انجام میده تا ما را به حداقل برسونه. پس ما حداقل چیزی که حریف میزاره برای ما باقی بمونه را در نظر میگیریم.

```
File Edit Selection Find View Goto Tools Project Preferences Help
optimal_strategy.py x untitled x
1 def optimal_strategy(A):
2     n = len(A)
3     dp = [[0] * n for _ in range(n)]
4
5     for gap in range(n):
6         for i in range(n - gap):
7             j = i + gap
8
9             x = dp[i+2][j] if i + 2 <= j else 0
10            y = dp[i+1][j-1] if i + 1 <= j - 1 else 0
11            z = dp[i][j-2] if i <= j - 2 else 0
12            #if mamadReza choose the right one
13            dp[i][j] = max(A[i] + min(x, y), A[j] + min(y, z))
14            #if mamadReza choose left one
15
16 total = sum(A)
17 mohamadreza = dp[0][n-1]
18 matin = total - mohamadreza
19
20 winner = "mohammadReza" if mohamadreza > matin else "matin"
21 print(winner)
22 print(f"{mohamadreza}, {matin}")
```

| i\j | 0 | 1 | 2 | 3 |
|-----|---|---|---|---|
| 0 | 4 | 6 | 6 | 9 |
| 1 | | 6 | 6 | 8 |
| 2 | | | 2 | 3 |
| 3 | | | | 3 |

این میشه مثالی از ورودی کوئرا که جواب در ایندکس $dp[0][n]$ است.

محمد کریمی ۴۰۲۳۶۱۳۰۶۰

امیررضا محمدی یگانه ۴۰۲۳۶۱۳۰۶۸