Dynamic Programming ---Coins in a line

Telescope Scheduling

**Coin in a line**

0/1 Knapsack

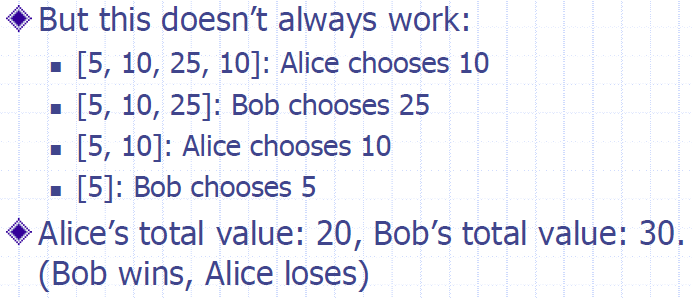
# How many coins must be in a line

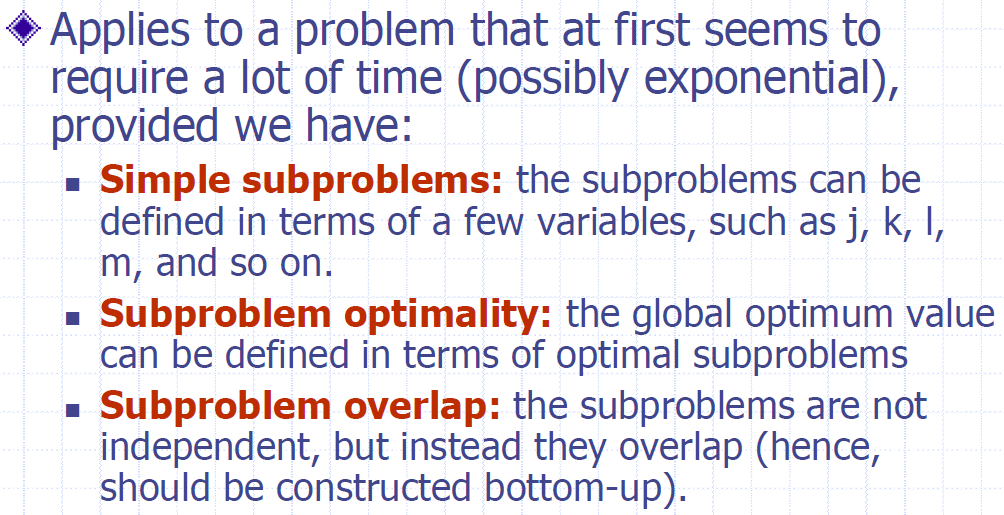
In this game, an **even number**, n, of coins, **of various face value**, are **placed in a line**.

# Does this require dynamic programming or greedy algorithm, explain and benefit of dynamic or greedy w.r.t. this problem

False Start 1: Greedy Method

A natural greedy strategy is “always choose the largest valued available coin.”





# Explain the problem and goal

In this game, an **even number**, n, of coins, **of various face value**, are **placed in a line**.

Alice and Bob are 2 players. They are required to choose coins only from leftist side or rightest side. Alice firstly chooses, try her best to do dynamic optimal strategy to make herself get maximum total value than Bob, and Bob will choose coin optimally to minimize what Alice can get next time.

The goal is to find an optimal selection strategy to let Alice get Maximum total value in this game than Bob.

# Explain the players strategies

Alice must choose based on the following reasoning:

**Case 1: Base case:**

If j = i + 1, then she should pick the larger of V[i] and V[j], and the game is over.



**Case 2:**

If Alice chooses coin i, then Bob will choose V[i + 1] or V[j], and he will do a strategy to minimize total coins value that Alice can get after. So in this case, Alice can get:



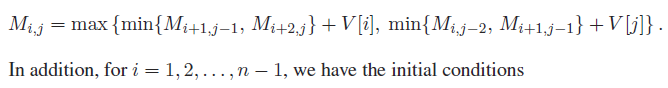
**Case 3:**

If Alice chooses coin j, then Bob will choose V[i] or V[j-1] ], and he will do a strategy to minimize total coins value that Alice can get after. So in this case, Alice can get:

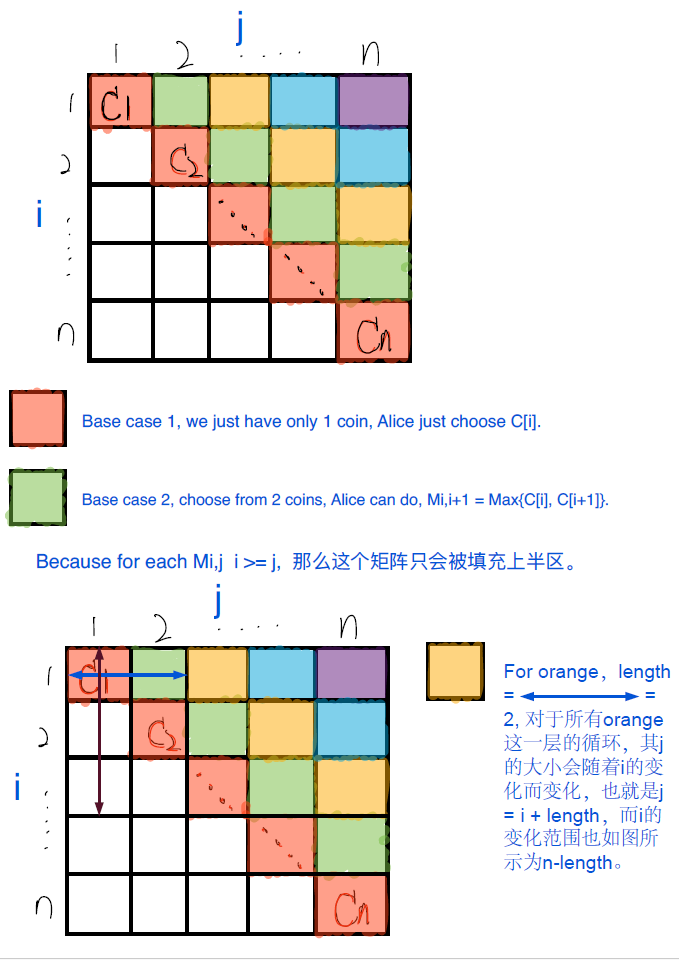


**So:**



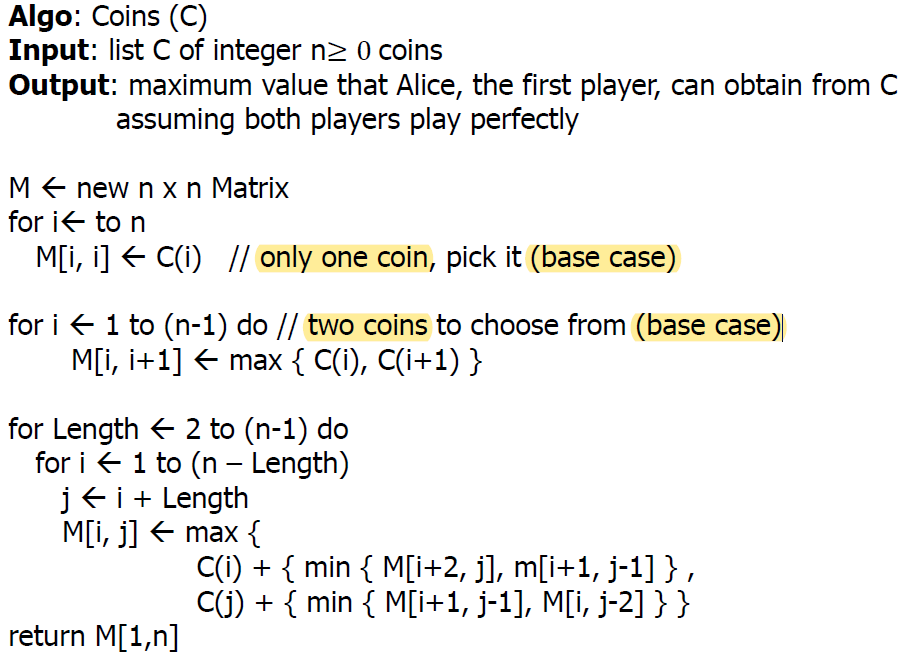


# Give examples of



# You do not have to memorize M(i,j) calculation, but do be able to explain the players strategies

# Given its code, explain it



# Algorithm runtime and explain it

Since there are O(n) iterations in this algorithm and each iteration runs in O(n) time, the total time for this algorithm is O(n^2).

# Greedy or dynamic and why

Dynamic programming is better than Greedy Algorithm for the "Coins in a Line" problem.

**Reason:**

DP can construct simple subproblems, create subproblem optimality, and solve the trouble of subproblem overlap. In this context, DP can globally explore all possible choices and consequences, ensuring a globally optimal solution.

While Greedy Algorithm seems faster, it may not guarantee global optimality, because it always does locally optimal choices without considering the global impact.

通用回答，仅需替换红色部分问题名字.