GeeksforGeeks

A computer science portal for geeks

GeeksQuiz

- Home
- Algorithms
- DS
- GATE
- Interview Corner
- Q&A
- C
- C++
- Java
- Books
- Contribute
- Ask a Q
- About

Array
Bit Magic
C/C++
Articles
GFacts
Linked List
MCQ
Misc
Output
String
Tree
Graph

Dynamic Programming | Set 31 (Optimal Strategy for a Game)

Problem statement: Consider a row of n coins of values v1 . . . vn, where n is even. We play a game against an opponent by alternating turns. In each turn, a player selects either the first or last coin from the row, removes it from the row permanently, and receives the value of the coin. Determine the maximum possible amount of money we can definitely win if we move first.

Note: The opponent is as clever as the user.

Let us understand the problem with few examples:

1. 5, 3, 7, 10 : The user collects maximum value as 15(10 + 5)

2. 8, 15, 3, 7 : The user collects maximum value as 22(7 + 15)

Does choosing the best at each move give an optimal solution?

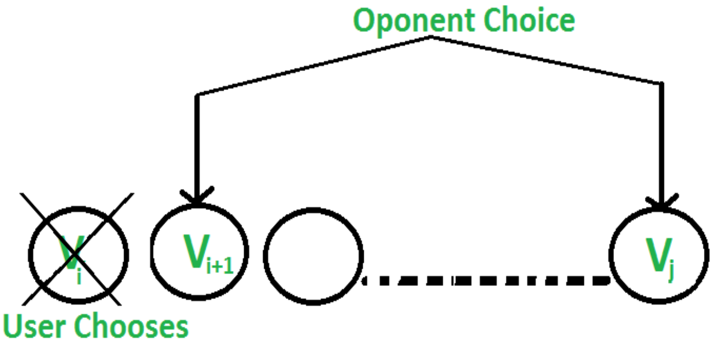
No. In the second example, this is how the game can finish:

1.
…….User chooses 8.
…….Opponent chooses 15.
…….User chooses 7.
…….Opponent chooses 3.
Total value collected by user is 15(8 + 7)

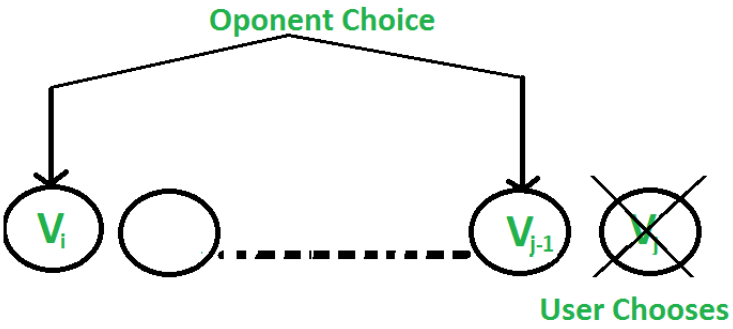
2.
…….User chooses 7.
…….Opponent chooses 8.
…….User chooses 15.
…….Opponent chooses 3.
Total value collected by user is 22(7 + 15)

So if the user follows the second game state, maximum value can be collected although the first move is not the best.

There are two choices:
1. The user chooses the ith coin with value Vi: The opponent either chooses (i+1)th coin or jth coin. The opponent intends to choose the coin which leaves the user with minimum value.
i.e. The user can collect the value Vi + min(F(i+2, j), F(i+1, j-1))



2. The user chooses the jth coin with value Vj: The opponent either chooses ith coin or (j-1)th coin. The opponent intends to choose the coin which leaves the user with minimum value.
i.e. The user can collect the value Vj + min(F(i+1, j-1), F(i, j-2))



Following is recursive solution that is based on above two choices. We take the maximum of two choices.

```
F(i, j)  represents the maximum value the user can collect from 
         i'th coin to j'th coin.

    F(i, j)  = Max(Vi + min(F(i+2, j), F(i+1, j-1) ), 
                   Vj + min(F(i+1, j-1), F(i, j-2) )) 
Base Cases
    F(i, j)  = Vi           If j == i
    F(i, j)  = max(Vi, Vj)  If j == i+1
```

Why Dynamic Programming?
The above relation exhibits overlapping sub-problems. In the above relation, F(i+1, j-1) is calculated twice.

```
// C program to find out maximum value from a given sequence of coins
#include <stdio.h>
#include <limits.h>

// Utility functions to get maximum and minimum of two intgers
int max(int a, int b)  {    return a > b ? a : b;  }
int min(int a, int b)  {    return a < b ? a : b;  }

// Returns optimal value possible that a player can collect from
// an array of coins of size n. Note than n must be even
int optimalStrategyOfGame(int* arr, int n)
{
    // Create a table to store solutions of subproblems
    int table[n][n], gap, i, j, x, y, z;

    // Fill table using above recursive formula. Note that the table
    // is filled in diagonal fashion (similar to http://goo.gl/PQqoS),
    // from diagonal elements to table[0][n-1] which is the result.
    for (gap = 0; gap < n; ++gap)
    {
        for (i = 0, j = gap; j < n; ++i, ++j)
        {
            // Here x is value of F(i+2, j), y is F(i+1, j-1) and
            // z is F(i, j-2) in above recursive formula
            x = ((i+2) <= j) ? table[i+2][j] : 0;
            y = ((i+1) <= (j-1)) ? table[i+1][j-1] : 0;
            z = (i <= (j-2))? table[i][j-2]: 0;

            table[i][j] = max(arr[i] + min(x, y), arr[j] + min(y, z));
        }
    }

    return table[0][n-1];
}

// Driver program to test above function
int main()
{
    int arr1[] = {8, 15, 3, 7};
    int n = sizeof(arr1)/sizeof(arr1[0]);
    printf("%d\n", optimalStrategyOfGame(arr1, n));
    int arr2[] = {2, 2, 2, 2};
    n = sizeof(arr2)/sizeof(arr2[0]);
    printf("%d\n", optimalStrategyOfGame(arr2, n));

    int arr3[] = {20, 30, 2, 2, 2, 10};
    n = sizeof(arr3)/sizeof(arr3[0]);
    printf("%d\n", optimalStrategyOfGame(arr3, n));

    return 0;
}
```

Output:

```
22
4
42
```

Exercise
Your thoughts on the strategy when the user wishes to only win instead of winning with the maximum value. Like above problem, number of coins is even.
Can Greedy approach work quite well and give an optimal solution? Will your answer change if number of coins is odd?

This article is compiled by Aashish Barnwal. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

Related Topics:

- Linearity of Expectation
- Iterative Tower of Hanoi
- Count possible ways to construct buildings
- Build Lowest Number by Removing n digits from a given number
- Set Cover Problem | Set 1 (Greedy Approximate Algorithm)
- Find number of days between two given dates
- How to print maximum number of A’s using given four keys
- Write an iterative O(Log y) function for pow(x, y)

Tags: Dynamic Programming

Tweet 3 g+1 3

Writing code in comment? Please use **ideone.com** and share the link here.

58 Comments GeeksforGeeks 1 Login

Recommend 1 Share Sort by Newest

Join the discussion…

Deepesh Maheshwari • 20 days ago

Hi,

When this condition occurs.I am able to visualize this one as user is heading with even steps

Base Cases
F(i, j) = Vi If j == i

^ | ⌄ • Reply • Share ›

Guest • 23 days ago

what is the time complexity of the above code ?

^ | ⌄ • Reply • Share ›

Goku ➦ Guest • 21 days ago

O(N^2)

^ | ⌄ • Reply • Share ›

Baybay • 3 months ago

There is a more elegant relationship.
F(i, j) = cumsum(i, j) - min(F(i+1, j) , F(i, j-1))
Cumsum(i,j): Sum of all the elements from i to j.

Reasoning: Once you select vi or vj, then your opponent will drive the game.

^ | ⌄ • Reply • Share ›

BayBay ➦ Baybay • 3 months ago

Here is the code:

```
int get_best_sum_alternating(vector<int> &num){

int N = num.size();

vector<int> cumsum(N,0);

cumsum[0] = num[0];
for (int i = 1; i < N; i++ ) {
cumsum[i] = (cumsum[i-1] + num[i]);
}

int F[N][N];

for (int gap = 0; gap < N; gap++){

for (int j = gap; j < N; j++){

int i = j - gap;
```

see more

⌃ | ⌄ • Reply • Share ›

vinaygaba • 3 months ago

Can someone please explain the logic behing these conditions?

```
x = ((i+2) <= j) ? table[i+2][j] : 0;
y = ((i+1) <= (j-1)) ? table[i+1][j-1] : 0;
z = (i <= (j-2))? table[i][j-2]: 0;
```

⌃ | ⌄ • Reply • Share ›

prashant jha • 4 months ago

http://ideone.com/X4HUFf

⌃ | ⌄ • Reply • Share ›

John • 4 months ago

Why doesn't this solution work for odd number of coins?

⌃ | ⌄ • Reply • Share ›

Chao Zhou • 6 months ago

There is a O(n^2) DP solution which can handle both even and odd number of coin cases, but it will need more space to store the intermediate result.

Here is the code in python:
http://ideone.com/3JZdjr

1 ⌃ | ⌄ • Reply • Share ›

guest • 7 months ago

can anybody provide the link of the realted ques on any of the online judges

⌃ | ⌄ • Reply • Share ›

Goku ➦ guest • 4 months ago

http://www.spoj.com/problems/T...

⌃ | ⌄ • Reply • Share ›

Ankush • 7 months ago

Add the values of odd index positions, call it sumOdd and similarly calculate sumEven. Choose to expose according to problem specification of minimum or maximum

⌃ | ⌄ • Reply • Share ›

rumpelstilzchen • 8 months ago

it is not necessary to store the whole matrix to get the solution. to calculate a column, u just need the two columns left from the one u want to calc. and since u need the corresponding value in the first of the two left columns just once, u can save the new value in its place. like that u just need to store 2 * size_sequence values. in this example i calculate from left to right and from bottom to top. u can also use two diagonals inplace of the two left col and then calc diagonal after diagonal. in case storage size matters

```
public static long findMax(){
// sequence is the array where ur sequence(of in this case coins) is stored
long x, y, z;
int i, j;
long[] col_1 = new long[n];
long[] col_2 = new long[n];
for(i = 0; i < n; i++){
for(j = i; j >= 0; j--){
boolean bool = i % 2 != 1;
x = j + 2 < n ? (bool ? col_1[j + 2] : col_2[j + 2]) : 0;
y = j + 1 < n ? (bool ? col_2[j + 1] : col_1[j + 1]) : 0;
```

see more

• Reply • Share ›

rumpelstilzchen rumpelstilzchen • 8 months ago

and if u calc diagonal after diagonal, i think u just need to store 6 values, independent from the sequence size

• Reply • Share ›

rumpelstilzchen rumpelstilzchen • 8 months ago

well the last if block should be:

```
if(bool){
col_1[j] = foo;
}else col_2[j] = foo;
```

• Reply • Share ›

dev21 • 9 months ago

Please tell if the following solution is also a DP one...

http://ideone.com/yN1g4f

• Reply • Share ›

pawan • 10 months ago

great solution

1 • Reply • Share ›

krishna • 10 months ago

the conditions,(i+2)<=j ,(i+1)<=(j-1),i<=(j-2) all are mathematically equivalent.

why don't we just put one if(i+2<=j){x=;,y=;,z=;} condition
correct me if i am wrong?

2 ⌃ | ⌄ • Reply • Share ›

Laxman ➦ krishna • 8 months ago

Substitute i = 0 and j = 0; They are not the same.

⌃ | ⌄ • Reply • Share ›

hesham • 10 months ago

What if the player does not have to select from either ends? Would someone please help how would it be solved in this case ? Thanks.

⌃ | ⌄ • Reply • Share ›

CS ➦ hesham • 10 months ago

Then the player just picks the maximum coin that is available

1 ⌃ | ⌄ • Reply • Share ›

lol ➦ CS • 8 months ago

haha

1 ⌃ | ⌄ • Reply • Share ›

shine • 10 months ago

If we are talking about winning and not just maximizing then what will happen in scenario of even n?
Plz don't try any hunch..
Give ur argument on the basis of any example or any mathematical proof..
I have already sorted out for n odd.

⌃ | ⌄ • Reply • Share ›

Aditya Joshi ➦ shine • 10 months ago

If we want to win we would want to get the maximum amount possible. Which is what we are doing anyway.

⌃ | ⌄ • Reply • Share ›

Amit • 10 months ago

Superb solution...!!!

⌃ | ⌄ • Reply • Share ›

Karshit Jaiswal • 10 months ago

Epic Explanation...!!
Thanks guys.. :)

1 ⌃ | ⌄ • Reply • Share ›

Arnab • a year ago

Arnab • a year ago

Please cooment on my code... which so far passed all tests.

complexity:O(n^2)

```
class optGameStgr{

public static void main(String[] args){

int[] coins={20, 30, 2, 2, 2, 10};

int n=coins.length;

int[][] gain=new int[n][n];

for(int[] row: gain)

Arrays.fill(row,0);

for(int l=1;l<=n;l++)

{
```

see more

1 ^ | ∨ • Reply • Share ›

NB • a year ago

Why should we assume that the opponent always chooses the maximum value? From his perspective, there may be a case when he chooses a smaller value that he believes leads to a better result.

Refer :
"Does choosing the best at each move give an optimal solution?"

^ | ∨ • Reply • Share ›

Karshit Jaiswal ➦ NB • 10 months ago

"The opponent intends to choose the coin which leaves the user with minimum value." refer explanation.

^ | ∨ • Reply • Share ›

Aenab ➦ Karshit Jaiswal • 6 months ago

We have to believe both are playing optimally

1 ^ | ∨ • Reply • Share ›

vrg • a year ago

Its very difficult to understand DP. How do you get intuition for using the table. Then filling it diagonally ?

How will you interpret the meaning of the table?

^ | ⌄ • Reply • Share ›

Arnab ➦ vrg • a year ago

Hey,
Dont't ever try to understand the DP problem in term of table. Try to break the problem down into smaller optimal substructure as u do in recursion/divide and conquer.Then try to formulate it bottom-up manner or top down using memoization. Intution of table will come automatically with practices.
Best wishes,

5 ^ | ⌄ • Reply • Share ›

Kang Zheng • a year ago

Doesn't the gap have to be even?

^ | ⌄ • Reply • Share ›

SDG • a year ago

```
/* Using Memoization Technique */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define min(a,b) ((a) > (b) ? (b) : (a))
#define max(a,b) ((a) > (b) ? (a) : (b))

int **dp;

int optimalStrategyOfGame(int arr[], int left, int right) {

int ret1, ret2;

if(right - left == 1) {
dp[left][right] = max(arr[left],arr[right]);
return max(arr[left],arr[right]);
}

if(dp[left][right] > 0 && left < right) {
```

see more

^ | ⌄ • Reply • Share ›

prashant • a year ago

```
/*

int fun(int arr[],int low,int high)

{
```

```
{

if(high-low==1)

{

if(arr[low]>arr[high])

return arr[low];

return arr[high];

}

return max((arr[low]+min(fun(arr,low+2,high),fun(arr,low+1,high-1))),

(arr[high]+min(fun(arr,low,high-2),fun(arr,low+1,high-1))));

}
*/
```

⌃ | ⌄ • Reply • Share ›

prashant • a year ago

naive recursice i

http://ideone.com/S6P9eP

⌃ | ⌄ • Reply • Share ›

prashant • a year ago

```
/*

int fun(int arr[],int low,int high,int &n)

{

int k1,k2,n1,p1,n2;

if(high-low==1)

{

if(arr[low]>arr[high])

{

n=arr[low];

return arr[high];

}
```

see more

• Reply • Share ›

rahul23 • 2 years ago

Does this also guarntee that user will win also?? i mean maximum value that user can made is the winning amount also?or there can be a case that maximum value obtained by the user made user to lose the game ..please tell

• Reply • Share ›

Karshit Jaiswal rahul23 • 10 months ago

Following this algorithm if you start first, you win.. !!

• Reply • Share ›

shine Karshit Jaiswal • 10 months ago

How can we say that?
if n is not even then first user may not be the winner
e.g
8,15,3
By above gaming strategy user1 can score maximum 11

• Reply • Share ›

Karshit Jaiswal shine • 10 months ago

We assume everyone get equal chances to play the game (i.e n is even) read the problem statement carefully.

• Reply • Share ›

shine Karshit Jaiswal • 10 months ago

Yes..I know n is even for the above problem statement but how can we say that maximum score for user1 will be winning score too...I am not able to prove it

• Reply • Share ›

Karshit Jaiswal shine • 10 months ago

Use Induction for even n to prove it.
Its easy.

• Reply • Share ›

bhavneet • 2 years ago

maxProfit(i, j) = maxprofit A can make from arrray i to j
maxProfit(i, j) = max(sum(i, j)-maxProfit(i+1, j) , sum(i, j)-maxProfit(i, j-1))
maxProfit(i, j) for i==j = arr[i]

sum(i, j) can be solved in logn time using sementtree so it is o(n2logn)

• Reply • Share ›

Spandan Pathak • 2 years ago

What role does constraint ," n is even ", has to play here ?

• Reply • Share ›

Guest ➦ Spandan Pathak • 2 years ago

Both users get equal number of coins.

3 • Reply • Share ›

Mukit Chowdhury • 2 years ago

My code is like below:
Is there any wrong??? If it is, then for which case ??

```
 #include<stdio.h>
#include<iostream>
#include<math.h>
#include<string.h>
#include<string>
#include<stdlib.h>
#include<map>
#include<vector>
#include<queue>
#include<stack>
#include<algorithm>
#include<set>
using namespace std;

// Define Some Variables
```

see more

• Reply • Share ›

Born Actor • 2 years ago

```
 #include <iostream>
#include<string>
#include<sstream>
#include<iomanip>
# include <stdio.h>
# include <math.h>
#include <vector>
#include <stdlib.h>
using namespace std;
int a[50][50];
int n;
int b[50];
int function(int begin, int end);
int max(int a, int b);
int min(int a, int b);
void print();
int main()
{
```

see more

^ | ⌄ • Reply • Share ›

Novice • 2 years ago

Hello,

WHY IT IS COMPARED LIKE BELOW:
x = ((i+2) <= j) ? table[i+2][j] : 0;
y = ((i+1) <= (j-1)) ? table[i+1][j-1] : 0;
z = (i <= (j-2))? table[i][j-2]: 0;

I AM NOT ABLE TO UNDERSTAND, ABOVE CHECKED CONDITIONS.

8 ^ | ⌄ • Reply • Share ›

prashant saxena ↪ Novice • a year ago

#include <iostream>

int max_t(int a, int b)

{

return (a>b)?a:b;

}

int min(int a, int b)

{

return (a<b)?a:b; }="" int="" max_profit(int="" v[],="" int="" begin,="" int="" end,="" int**="" table)="" {="" if(begin="=" end)="" return="" v[begin];="" if(begin+1="=" end)="" return="" max_t(v[begin],="" v[end]);="" if(*(*(table+begin)+="" end)="" !="-1)" return="" *(*(table+begin)+end);="" *(*(table+begin)+="" end)="max_t(" v[begin]+min(="" max_profit(v,="" begin+1,="" end-1,="" table),="" max_profit(v,="" begin+2,="" end,="" table)),="" v[end]+min(max_profit(v,="" begin,end-2,="" table),="" max_profit(v,=""

see more

^ | ⌄ • Reply • Share ›

Reply Share

Load more comments

Subscribe Add Disqus to your site Privacy

-
-
-
-
    - Interview Experiences
    - Advanced Data Structures
    - Dynamic Programming
    - Greedy Algorithms
    - Backtracking
    - Pattern Searching
    - Divide & Conquer
    - Mathematical Algorithms
    - Recursion
    - Geometric Algorithms
-
- ## Popular Posts
    - All permutations of a given string
    - Memory Layout of C Programs
    - Understanding “extern” keyword in C
    - Median of two sorted arrays
    - Tree traversal without recursion and without stack!
    - Structure Member Alignment, Padding and Data Packing
    - Intersection point of two Linked Lists
    - Lowest Common Ancestor in a BST.
    - Check if a binary tree is BST or not
    - Sorted Linked List to Balanced BST
- Follow @GeeksforGeeks

- **Recent Comments**

  - lt_k

    i need help for coding this function in java...

    Java Programming Language · 2 hours ago

  - Piyush

    What is the purpose of else if (recStack[*i])...

    Detect Cycle in a Directed Graph · 2 hours ago

  - Andy Toh

    My compile-time solution, which agrees with the...

    Dynamic Programming | Set 16 (Floyd Warshall Algorithm) · 2 hours ago

  - lucy

    because we first fill zero in first col and...

    Dynamic Programming | Set 29 (Longest Common Substring) · 2 hours ago

  - lucy

    @GeeksforGeeks i don't n know what is this long...

    Dynamic Programming | Set 28 (Minimum insertions to form a palindrome) · 3 hours ago

  - manish

    Because TAN is not a subsequence of RANT. ANT...

    Given two strings, find if first string is a subsequence of second · 3 hours ago

-