

1620 - problem solving

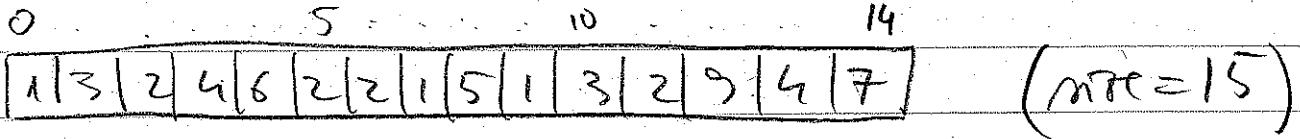
Nov 27, 2009

Assignment 8

- Problem: • array a of max 1000 entries is given. It contains 'nre' positive values representing cost.
- Integer budget is given.

Task

Print out all "maximal" array ranges within budget.
↓
useful.



Budget = 6

(A maximal range = one that cannot grow left or right)

Task 1

- give one maximal range. ex: (7, 8)
- how did you obtain it in your head?

(9, 11)

Method: Start @ some index i

{ add entries to the right until the sum exceeds budget}

What input do you need to complete the operation?

- a) array
 - b) index (leftmost)
 - c) budget
- } candidates for parameter list.

What is the output?

- right index,

What is a good name for the operation?

int maximalRange (int a[], int n, int left, int budget);

Observation What do you notice about the plan (a) to (c)) and the prototype of maximalRange ()?

array → int a[]
int n

{}

the way to work with

arrays in C++

(a detail)

We could easily mistake & forget n in our translation plan → C++.

Solution

const int MAXSIZE = 100;

- structures:

```
struct Array {  
    int a[MAXSIZE];  
    int max;};  
}
```

Advantage → we don't forget to pass 'arr' as argument to functions working with the array.
→ we don't forget which variable we're using for the size of an array if we have several arrays.

Obs → Array is now a structure & it is passed by value or by reference according to 'struct' rules.

int maximalRange (const Array& a, int left,
 int budget)

// returns the right index for maximal range.

What should the output be for maximalRange (a, 12, 6)? (no range!)

maximal Range (cost array & a, int left, int budget)
// returns right index OR
-1 if no range.

Homework Implement MaximalRange().

Plan • compute the Σ of entries
starting @ index, to the right
• stop right before the $\Sigma >$ budget.

Back to task, is now printing all maximal
array ranges within budget easy?

Consider:

```
for (left = 0; left < a.size(); left++)  
    cout << " " << left << " " <<  
        maximalRange(a, left, budget)  
    << " ");
```

Now you have a method to list all maximal
ranges within budget for the cost array.
Use it to compute total profits on the profit
array, by picking the sum of profits
between left & maximalRange(a, left, budget).

Keep the range that gives maximum profit.

Homework:

- write a function that gives the \sum of elements between a left index & a right index.
If right index < left index, 0 is returned.
- use this function & maximalRange() in the main() routine for your program for problem 3 in Assignment 8.

Good Luck.