1620 – problem solving. Nov 27, 2009

Assignment 8

Problem: • array a of max 1000 entries is given. It contains 'size' positive values representing cost.

• Integer budget is given.

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

0					5					10				14
1	3	2	4	6	2	2	1	5	1	3	2	3	4	7

(size = 15)

Budget = 6

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

Task 1 – give one maximal range. ex: (7,8) (9,11)
– how did you obtain it in your head?

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 (left/most) } candidates for parameter list.
c) budget

What is the output?
- right index.

What is a good name for the operation?

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

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

array → int a[]
int size

the way to work with arrays in C++ (a detail)

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

Solution

const int MAXSIZE = 1000;

- structures!

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

Advantage → we don't forget to pass 'size' 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!)

maximalRange(const 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 left to the right
- stop right before the Σ > 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 finding the sum of profits between left & maximalRange(a, left, budget).

Keep the range that gives maximum profit.

Homework:

- write a function that gives the Σ 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.