CS110: Computer Programming Lab
Department of CSE
IIT, Guwahati

Module 04 Stage 01 Exercise 13

Exercise

Mr and Mrs A are about to separate. They count all their money into bundles; these money bundles are of random values.

For ease, function `main()` has listed the values in an array (number are in crores of rupee).

```
int main()
{
    int a[] = {351, 400, 471, 850, 801, 500, 531, 300};};
    printf( "Best value %d\n", maxy (a, 0, 7));
}
```

They lay the bundles in a row and agree that they will take turn to pick one bundle each. They also agree that only a bundle at the end can be picked; one can pick bundle from any one of the two ends

Both are very prudent and wise persons. How much the person picking the first bundle get?

The answer to the described problem is 2154. However, if 850 bundle is changed to 1850 the answer becomes 3050.

Help and Advice

The problem may be solved through two recursive functions. Let us call them `maxy()` and `miny()`. The arguments in the function are array `a`, and two indices `left` and `right`. These are the two extreme points of the bundles not yet taken.

Function `maxy()` tries each of the two possible choices (left and right ends of the row of bundles) and determines which one is the better option to maximize the final/total gains in the current position.

Picking a smaller value bundle now may give better opportunities when the person gets the other choices later. Recall that the other person cannot choose a big value bundle if it is not at one of the ends. Thus `maxy()` returns the maximum value of the bundles that the caller can receive from the bundles in the row.

Function `miny()` returns the minimum amount that would be received. This is what is guaranteed irrespective of what other persons choose from the present state of the row.

The following identity is obvious:

`Total (a,from,to) == maxy (a,from,to) + miny (a,from,to)`
Where `Total` is total money in array from locations `from` to `to` (both included).

The idea is simple, after person P has chosen a bundle (from one end of the remaining row), function `miny()` provides the amount person can expect from the later pickings.

Recursive calls could be terminated when two or fewer bundles remain in the row. Decision is simple.

In the program, function `miny()` calls function `maxy()` twice. And, function `maxy()` calls function `miny()` twice. In each of these calls, one call is to determine the total collections possible when the "left" bundle is selected; and the other call is when the "right" bundle is selected. The called (recursive) function provides this value. Once total collections from the two alternate cases is known, the person selects the better option. Note, the definition of better in the two functions is contradictory. This is so as one function seeks to maximize and the other seeks to minimize the returned value.

Error Reporting and Suggestions for Improvements

My sincere apologies if the document has errors or mistakes. Please report errors in this document to vmm@iitg.ernet.in. Also, I welcome suggestions and advice to improve the quality of the document for the students of CS110.