CIS 621 Algorithms and Complexity
Winter 2016

Assignment 6

due Monday, February 29, 2016

We revisit the machine reliability problem from the DPV text and assignment 5.

problem recollection:
A mission-critical production system has n stages that have to be performed sequentially; stage i is performed by machine M_i. Each machine M_i has a probability r_i of functioning reliably and a probability $1 - r_i$ of failing (and the failures are independent). Therefore, if we implement each stage with a single machine, the probability that the whole system works is $r_1 \cdot r_2 \cdots r_n$. To improve this probability we add redundancy by having m_i copies of the machine M_i that performs stage i. The probability that all m_i copies fail simultaneously is only $(1 - r_i)^{m_i}$, so the probability that stage i is completed correctly is $1 - (1 - r_i)^{m_i}$ and the probability that the whole system works is $\Pi_{i=1}^{n}[1-(1-r_i)^{m_i}]$. Each machine M_i has a cost c_i, and there is a total budget B to buy machines. (Assume that B and the c_i are positive integers.)

Given the probabilities $r_1, r_2, \ldots, r_n$, the costs $c_1, c_2, \ldots, c_n$, and the budget B, find the maximum reliability that can be achieved within budget B.

write actual code now:
Implement a dynamic programming solution, preferably in *java* or *python*, to take a problem instance from standard input and determine the maximum reliability achievable.

- You should provide both an iterative and memoized solution.
- In addition to providing the maximum reliability, show how many machines of each type achieve that reliability bound within the budget.
- Perform the previous step for both the iterative and memoized version (usually the same procedure would work).
- Ideally, provide memoization statistics, showing how many of the array locations were filled in by the recursion.
- If you're using *java*, the final reliability may need to be a `double`.
- If you wish to use another language, please check with the instructor.

input format:
The input will be a text file, to be read from standard input. For example, if the input file is `inSample.txt` and the java class is `reliability`, the command to run it will be `java reliability < inSample.txt`. For python, we would say `python reliability.py < inSample.txt`.

In the text file, the first line will be an integer `B`, the budget. The second line will be an integer `N`, the number of machines. One or the other may be zero, but neither will be negative. What follows

will be `N` lines of the form `C R`, where `C` is an integer (cost) and `R` is a float (reliability). You may assume that there are exactly `N` lines followed by a blank line, $\texttt{C} \geq 1$, and $0 < \texttt{R} \leq 1$.

sample input:

```
7500
12
12 0.25
10 0.35
15 0.2
5 0.3
20 0.4
17 0.3
10 0.4
15 0.3
13 0.2
9 0.15
16 0.1
19 0.13
```

sample output:

```
Budget: 7500
Number machines: 12

Iterated Version:
Maximum reliability: 0.99979661309622
73 copies of machine 12 of cost 19
95 copies of machine 11 of cost 16
69 copies of machine 10 of cost 9
49 copies of machine 9 of cost 13
32 copies of machine 8 of cost 15
24 copies of machine 7 of cost 10
31 copies of machine 6 of cost 17
22 copies of machine 5 of cost 20
35 copies of machine 4 of cost 5
49 copies of machine 3 of cost 15
27 copies of machine 2 of cost 10
39 copies of machine 1 of cost 12

Memoized Version:
Maximum reliability: 0.99979661309622
73 copies of machine 12 of cost 19
95 copies of machine 11 of cost 16
```

```
69 copies of machine 10 of cost 9
49 copies of machine 9 of cost 13
32 copies of machine 8 of cost 15
24 copies of machine 7 of cost 10
31 copies of machine 6 of cost 17
22 copies of machine 5 of cost 20
35 copies of machine 4 of cost 5
49 copies of machine 3 of cost 15
27 copies of machine 2 of cost 10
39 copies of machine 1 of cost 12

Memoization Statistics:
Total locations: 90000
Number used: 74210
Percentage used: 82.45556
```

what to submit

Send an email to the instructor containing your *.java* or *.py* file (hopefully a single file) by midnight of the due date. Also submit a hard-copy of your code - this may be handed in later. You may also wish to provide a document with any comments, observations, or execution instructions (but still expect it to be run at the command line as described above).