CISC-365
2009
Lab # 9
Week of November 9

As happens every year about this time, waves of Zombies are approaching Queen's University, eager to feast on the brains of bright young students and even dim old professors. This year things are going to be different – the AMS has hired you to organize the defenses, and you have at your disposal a prototype ZombieZapper.*

The ZombieZapper is a contraption that attracts and fries zombies. It is 100% effective on zombies that it affects, but it has some limitations:

- it can only be fired once per day
- the number of zombies it can affect is a function of the number of days the ZZ has been allowed to charge since it was last fired. Specifically, the number of zombies the ZZ can affect is 1 on the first day after firing, 2 the next day, 4 the next, and so on up to a maximum of 32 (this is the maximum battery capacity of the ZZ).
- the ZZ starts out completely drained – if you fire it on Day 0, you will only be able to fry one zombie.

Any zombie not zapped on the day it arrives is free to pillage Queen's to its undead heart's delight – they cannot be zapped on later days. So if the ZZ is fired on day x and it has been charging for c days, the number of zombies fried will be

min{2^(c-1), 32, number of zombies arriving on day x}

Fortunately, the approaching groups of zombies have been located and counted by intrepid scouts sent out by the School of Computing so you have complete information regarding how many zombies will arrive each day. Your task is to decide which days to fire the ZZ so as to maximize the number of zombies eliminated.

Example of the information:

Day	0	1	2	3	4
Zombies arriving	5	4	5	3	4

In the example above, one optimal solution is to fire the ZZ on Day 2 (frying 4 zombies) and Day 4 (frying 2 zombies). There is at least one other optimal solution that also fries 6 zombies.

Decide whether this problem is best solved by a Dynamic Programming approach or a Greedy approach. Implement your solution.

If you get totally stuck on this point, see Part 2.

Input for this lab consists of a text file containing a number of instances of the problem. The first line of the file contains an integer that specifies the number of instances to be solved. Each instance is defined by two lines of the file. The first line for an instance contains an integer that specifies the number of days in this instance. The second line for an instance gives the number of zombies for each day in this instance.

For example, the input file might look like

```
2
5
12      8       7       2       4
4
1       7       4       3
```

---

* This problem is loosely based on a problem found in Kleinberg and Tardos's book "Algorithm Design"