

Homework #3 Solutions

Raja Kantheti

Gadget Testing

A company wants to determine the highest floor of its n -story headquarters from which a gadget can fall without breaking. The company only has one gadget to test with, and if the gadget breaks, it cannot be repaired — that is, by the time the gadget breaks, you better have the answer to your question.

1. (1 gadget)

Problem: Design an algorithm to solve this problem. Provide pseudocode or a precise description of your approach. Determine the best and worst-case number of times you need to drop the gadget to solve the problem.

Algorithm:

1. Start from the first floor and drop the gadget.
2. Move to the next floor and repeat the process.
3. Continue this until the gadget breaks.

Pseudocode:

```
function find_breaking_floor(n):  
    for i from 1 to n:  
        if gadget breaks when dropped from floor i:  
            return i - 1 # The highest safe floor  
    return n # Gadget did not break from any floor
```

Best-Case and Worst-Case Analysis:

- **Best-Case:** The gadget breaks on the first drop (1 drop).
- **Worst-Case:** The gadget breaks on the n th drop (n drops).

2. (uniform average case)

Problem: Suppose that it is equally likely that the gadget will break from any floor. If you perform this experiment many times, what is the average number of times you need to drop the gadget?

Solution: If each floor has an equal probability of being the breaking floor:

$$p_i = \frac{1}{n}$$

The expected number of drops, E , is:

$$E = \sum_{i=1}^n i \cdot p_i = \sum_{i=1}^n i \cdot \frac{1}{n} = \frac{1}{n} \sum_{i=1}^n i = \frac{1}{n} \cdot \frac{n(n+1)}{2} = \frac{n+1}{2}$$

So, the average number of drops needed is $\frac{n+1}{2}$.

3. (skewed average case)

Problem: Suppose instead that the probability of breaking when the gadget falls from floor k is proportional to $\frac{1}{k}$. What is the order of growth (in terms of $\Theta(\cdot)$ and n) of the average number of gadget drops in this case?

Solution: if the probability of breaking on floor k is proportional to $\frac{1}{k}$:

$$p_k = \frac{M}{k}$$

where M is just some constant.

For p_k to be a valid probability distribution, they should all give the sum of 1.

$$\sum_{k=1}^n \frac{M}{k} = 1 \implies M = \frac{1}{H_n}$$

where H_n is the n -th Harmonic number.

The expected number of drops:

$$E = \sum_{k=1}^n k \cdot \frac{1}{kH_n} = \frac{1}{H_n} \sum_{k=1}^n 1 = \frac{n}{H_n}$$

H_n grows as $\Theta(\log n)$, so $E = \Theta(n/\log n)$.

4. (2 gadgets)

Problem: Now, suppose that the company has two identical gadgets to experiment with. If one of the gadgets gets broken, it cannot be repaired, and the experiment will have to be completed with the remaining gadget. Use the second gadget to improve on your algorithm from part (a). That is, design an algorithm that has worst-case time complexity better than $\Theta(n)$. What is the order of growth of the worst-case number of drops for your redesigned algorithm?

Solution: The optimal strategy for minimizing the worst-case number of drops with two gadgets is to choose an interval size k such that the total number of drops is minimized. The interval k should be chosen as follows:

$$k = \lceil \sqrt{n} \rceil$$

Then, drop the first gadget from floors $k, 2k, 3k, \dots$ until it breaks. Suppose it breaks at floor mk . Then, use the second gadget to test each floor sequentially between $(m-1)k+1$ and $mk-1$.

Example:

1. Choose $k \approx \sqrt{n}$.
2. Drop the first gadget from floors $k, 2k, 3k, \dots$ until it breaks.
3. Use the second gadget to test each floor sequentially between the previous breaking point and the current breaking point.

Pseudocode:

```
function find_breaking_floor_2_gadgets(n):
    interval = ceil(sqrt(n))
    previous_floor = 0

    for i from interval to n by interval:
        if gadget breaks when dropped from floor i:
            for j from previous_floor + 1 to i:
                if gadget breaks when dropped from floor j:
                    return j - 1
            return i - 1
        previous_floor = i
    return n
```

Worst-Case Analysis:

- The interval is \sqrt{n} .
- The order of growth for worst case number of drops is $\Theta(\sqrt{n})$.

5. (the tradeoff between best and worst case)

Problem: What is the best-case number of drops for your 2-gadget algorithm? Discuss the relationship between this answer and the best-case answer you gave in part (1).

Solution:

- The best-case number of drops for the 2-gadget algorithm is 1 drop, which occurs if the gadget breaks on the first drop in the interval determination step.
- The best-case scenario does not change with two gadgets. However, the worst-case scenario improves significantly from n drops to $2\sqrt{n}$ drops.