

Analysis: Go, Gopher!

Test set 1 (Visible)

For test set 1, we need to prepare all the cells within a grid-aligned rectangle of size at least 20. Before starting to deploy the gopher, let's choose a rectangular target region of size at least 20. We will try to prepare all the cells within this target region. One option is to choose a 4 x 5 target region. We could have chosen 3 x 7, 5 x 5 etc, but it should not be too big. It does not matter where we place this target region in our initial 1000 x 1000 matrix. So let's place it such that one corner of the rectangle is at (1, 1) and the opposite corner is at (4, 5). Here (r, c) refers to the cell in the r-th row and the c-th column of the original matrix. Now the question is: can we come up with a strategy that will prepare all of the cells in this target region, and no other cells?

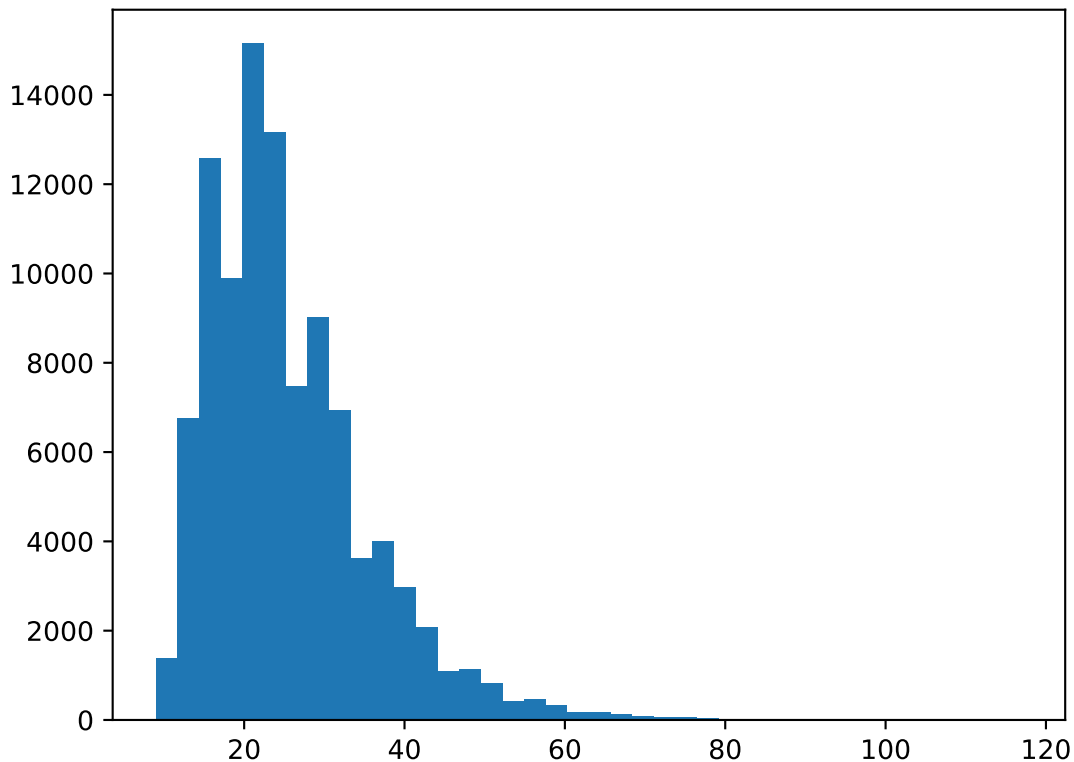
Let's visualize our 4 x 5 target region as follows (row and column numbers are given for convenience):

```
  12345
1  xxxxx
2  x@@@x
3  x@@@x
4  xxxxx
```

We marked the internal cells with @ and the border cells with x. The gopher should not be deployed on the border cells, because that might cause it to prepare a cell outside of our target region. We will only deploy on the internal cells (cells marked with @ in the above picture). We can deploy the gopher 1000 times in total and we have 6 internal cells. Let's deploy the gopher to each of the internal cells $\text{floor}(1000 / 6) = 166$ times. Will this be enough to solve test set 1?

To answer this crucial question, let us compute the probability that after deploying the gopher as described above, the (1, 1) cell is still unprepared. Notice that this cell can only be prepared by deploying the gopher at (2, 2). Every time we do this, it has a 1/9 probability of preparing (1, 1). Looking at it from another angle, every time we do this, it has an 8/9 probability of not preparing (1, 1). So the probability that (1, 1) will not be prepared after 166 deployments to (2, 2) is $(8/9)^{166} = 3.226 \times 10^{-9}$, which is quite small. Realistically, we do not need to worry about this happening to any of the four corners (since the probability of this happening to at least one of the four corners is $1 - (1 - 3.226 \times 10^{-9})^4 = 1.29 \times 10^{-8}$). Other cells are adjacent to more than one internal cell, and thus they are more likely to be prepared than the corners. So this solution is sufficient to pass test set 1.

To convince you further we ran a simulation. We deployed the gopher to the center of a 3 x 3 region until all the nine cells in the region is prepared. We ran this 100000 times and following is the result:



The above figure shows how many of our 100000 simulations (y axis) required each possible number of deployments (x axis). As we can see, out of 100000 simulations the maximum number of times we needed to deploy the gopher was not more than 120. So 166 deployments would be more than enough to prepare an internal cell and all of its surroundings. Moreover, once we have prepared an internal cell and all eight of its neighbors, there is no reason to deploy the gopher there again, so we will almost certainly have more than 166 deployments to use to fill in the last stubborn cell if necessary.

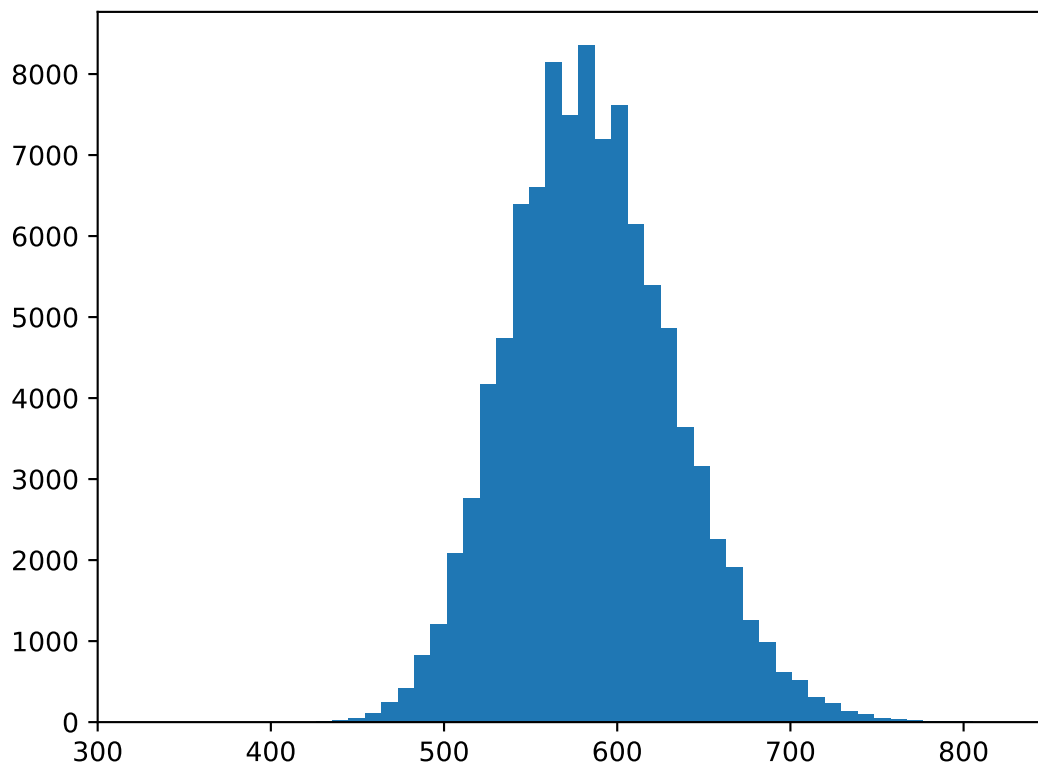
Test set 2 (Hidden)

Now we need to create a rectangular region of area at least 200. If we used the same strategy described above, for, say a 10 x 20 size rectangle, we could make $\text{floor}(1000 / (18 * 8)) = 6$ deployments to each of the internal cells. But then the probability of (1, 1) being not prepared would be: $(8/9)^6 = 0.49327$, which is way too high!

How can we improve this strategy? We can observe that most of the cells can be prepared from a number of locations. For example, the cell (2, 2) can potentially be prepared on a deployment to any of the internal cells around it, or to the cell itself. What if we divide our rectangular region into disjoint 3 x 3 regions and only deploy the gopher to the center cells of those regions? This way each of the cells can be prepared from only one cell. To sum things up, our plan is:

- Select a large enough rectangle, say 3 x 69.
- For convenience, we will place the corner of this 3 x 69 rectangle at (1, 1).
- We will divide our initial 3 x 69 region into $69/3 = 23$ disjoint 3 x 3 regions. That is, we will deploy the gopher only to (2, 2), (2, 5), (2, 8) ... (2, 68): the center cells of those regions.
- We will keep deploying the gopher at (2, 2) until all the cells inside the 3 x 3 grid centered on (2, 2) are prepared.
- Then we will deploy at (2, 5) and so on.

Is that enough? As the above simulation showed, sometimes it requires 120 deployments to prepare entire 3 x 3. So in the worst case $23 \times 120 = 2760$ deployments to prepare entire 3 x 69 region which is more than our limit 1000. However, such worst case will not happen always. We ran another simulation to examine our new strategy:



The above figure shows how many of our 100000 simulations (y axis) required each possible number of deployments (x axis) to prepare all the cells inside 3 x 69 target region. As we can see, the maximum we needed is no more than 850 and our limit was 1000. So we can be confident that this strategy is good enough to pass test set 2.

There are many other strategies. One may be, to deploy the gopher to the cell that has the largest number of unprepared cells in 3 x 3 region centering at that cell. This strategy yields following simulation result (100000 runs), which looks better than the one before:

