

Algorithmic Methods of Data Mining - Assignment 2

Maksad Donayorov

October 16, 2018

1. **Problem:** *Bag of words:*

2. **Problem:** $O(\log w \log n)$:

Before analyzing the complexity and responding to the question it could be nice to see an algorithm that finds the *max* number in an array. Such algorithm can be easily written in python, that gives us a clear picture of how the computation is done:

```
1 def get_max(X):
2     max_num = -1
3     for i in X:
4         if max_num < i:
5             max_num = i
6     return max_num
```

The best case for finding the *max* element is when the first element is bigger than all of the other elements of X , which has complexity of $O(1)$. And the worst case is when the last element is the largest and all the consecutive numbers proceed from small to large. This case has a complexity of $O(n)$. The tricky part is the average case, where we have to compute the complexity for *the number of times that max is assigned to an element*. To do that, we need to do a probabilistic analysis for the average case of assigning *max* to X_i that will be made on execution.

Assuming that $X[1, 2, 3 \dots m]$ is drawn independently and uniformly at random from the interval $(0, 1)$, the expected probability will be:

$$E[x] = \sum_{i=1}^m Pr(X_i)$$

Where $Pr(X_i)$ is the probability of the i_{th} element being the *max* element in $X[1, 2, 3 \dots m]$.

3. **Problem:** *Reservoir algorithm for sampling 1 element in a data stream:*

4. **Problem:** *Resort to sampling of good items:*