Assignment 1

Thomas Vinther & Jens Kristian Refsgaard Nielsen

04-09-18

Assume we have an infinite long sorted sequence of numbers $x_1 < x_2 < x_3 < \cdots < x_{d-1} < x_d < x_{d+1} < x_{d+2} < \cdots$ and we want to find the position of a number y in this list, i.e. we want to find the index d, such that $y = x_d$ if y is contained in the list, or otherwise if y is not in the list the successor of y in the list, e.g. $x_{d-1} < y <= x_d$ (here we assume $x_0 = -\infty$ if $y < x_1$).

Subtask 1:

We construct the following algorithm, given binary search from introduction to algorithms third edition:

```
public int binSearch(x,t,p,r){
    low = p;
    high = max(p,r+1);
    while (low < high) {
        mid = floor((low+high)/2);
        if (x ≤ T[mid]) {
            high=mid;
        }
        else {
            low = mid+1;
        }
    }
    return high;
}

public int infSearch(x,y){
    i = 1;
    n = max(floor(|y|),1000);
    if (y ≤ x[n]){
        return binSearch(y,x,1,n);
    {
    while (true) {
        if (x[in + 1] ≤ y ≤ x[n(i + 1)]) {
            return binSearch(y,x,in+1,n(i+1));
        }
        i++;
```

```
    }
}
```

Subtask 2:

We wish to show that we return the correct index d.

Now there are some special cases worth noting:
Special case 1: If the sequence x is bounded and real, it is also convergent. Note that in this case $\lim_{n\to\infty} x_n = \inf\{k \in \mathbb{R}_+ : \forall n \in \mathbb{N} : x_n \leq k\} = \sup\{x_n : n \in \mathbb{N}\} =: x_\infty$ because it is increasing. Now in this case our algorithm will never terminate if $x_\infty \leq y$. But when confronting Gerth, he said that we should assume that the sequence is unbounded.
Special case 2: If $\forall n \in \mathbb{N} : x_n \in \mathbb{N} \setminus \{0\}$ we can simply do a binary search on the subsequence $x_1, \ldots, x_{floor(|y|)}$ because in this case $\forall m \in \mathbb{N} : m \leq x_m$, and our algorithm solves this issue before entering the while loop. Due to our smart choice of n in the algorithm.
In any other case we divide our sequence into n sized parts, and run binary search on the subsequence $x_{in+1}, \ldots, x_{n(i+1)}$ for an i such that $x_{in+1} \leq y \leq x_{n(i+1)}$. Such an i exists because the sequence is unbounded. And we know from the lectures that binary search finds the index d that we are tasked to find.

Subtask 3:

We wish to determine how many comparisons the algoritm performs.

$$worstCase(d) = \begin{cases} \log_2(n) + c & \text{if } d \leq n \\ \log_2(n) + roof(\frac{d}{n}) + c & \text{if } d > n \end{cases}$$

We know that binary search in the worst case makes $\log_2(n)$ comparisons. And we look at d/n intervals before finding the correct interval to binary search.
In the best case $y = x_{floor(|y|)/2}$ and we are done in 2 comparisons, since binary search has a best case of O(1).