RWTH Aachen **SS 2012**
Lehrgebiet Theoretische Informatik Problem Set 7
Langer—Reidl—Rossmanith—Sikdar 4.6.2012

Analysis of Algorithms — Tutorial

Problem 7-1

Consider the following algorithm that searches an element x in a sorted array a of length $n = km + 1$:

```
i := 1;
while a[i] ≤ x
  if a[i] = x then return i;
  i := i + m;
  if i > n return 0;
for j = i − 1 downto max(1, i − (m − 1))
  if a[j] = x then return j;
  if a[j] < x then return 0;
return 0;
```

a) Draw the search tree and compute the internal and external path length for $n = 10$ and $m = 3$.

b) Determine C^+ and C^- for arbitrary m, k.

c) What is, for given n, the best choice for m w.r.t. the running time?

Problem 7-2

Consider the following two programs for searching elements in ordered arrays:

```
int binsearch(double v)
{
  int l, r, m;
  l = 1; r = n;
  while (l ≤ r) {
    m = (r + l)/2;
    if (v ≡ a[m]) return 1;
    if (v < a[m]) r = m − 1; else l = m + 1;
  }
  return 0;
}
```

```
int binsearch2(double v)
{
  int l, r, m;
  l = 1; r = n;
  while (r − l > 1) {
    m = (r + l)/2;
    if (v < a[m]) r = m − 1; else l = m;
  }
  if (a[l] ≡ v) return 1;
  if (a[r] ≡ v) return 1;
  return 0;
}
```

Determine the number of executions of **if**-statements in both problems when searching for an element v, in case of both, the successful and unsuccessful search.
Please give an exact solution for *binsearch* and an estimation of the form $f(n) + O(1)$ for for *binsearch2*.
Prerequisites:

- The array contains n *different* elements.
- For the successful search, each element is searched for with equal probability.
- For the unsuccessful search, v is chosen randomly, s.t., with probability $\frac{1}{n+1}$ is "in" one of the $n + 1$ possible gaps.

Homework Assignment 7-1 (10 Points)
Consider the following algorithms for searching an element x in an ordered array a of length n. Here, m, is some fixed, but known integer.
Draw the search tree and compute internal and external path lengths for $n = 17$ and $m = 3$.

Homework Assignment 7-2 (10 Points)
The external path length of a search tree for binary search — and thus the average number of comparisons – was given in the lecture, but its correctness has not been proved. Prove it!
Hint: Find a recurrence formula for the number of comparisons.