S. Schwoon / D. Suwimonteerabuth Winter Term 2008/09

Fundamental Algorithms

Solution Keys 4

1. (a) **Procedure** `selection`

Input: array a of length n, index s
Output: the s-th smallest element

$i := 1$; $j := n$;
while `true` **do**
  $k, l :=$ `partition`(a, i, j);
  if $s \leq k$ **then** $j := k$;
  else if $s \geq l$ **then** $i := l$;
  else return $a[s]$;
od

To ease the complexity analysis, we assume from now on that elements of a are distinct. The worse case happens when e.g. a is sorted and $s = n$. If this is the case, `selection` takes $\Theta(n^2)$ time.

For the average case, we need to consider all possible permutations of elements of a. Assume that all $n!$ permutations are equally likely. Let $t(n)$ be the average time for `selection`:

$$t(n) = g(n) + \frac{1}{n}\sum_{l=0}^{n-1} t(l) \leq dn + \frac{1}{n}\sum_{l=0}^{n-1} t(l) \ ,$$

where $g(n) \in \Theta(n)$ is the time for `partition` and d is a constant bound. We now prove that $t(n) \in \mathcal{O}(n)$ by showing that there exists a constant c such that $t(n) \leq cn$. We proceed by induction.

Basis $t(0) = 0$ and $t(1) = d$. Choose $c \geq d$.

Inductive step From the definition and the induction hypothesis

$$t(n) \leq dn + \frac{1}{n}\sum_{l=0}^{n-1} t(l) \leq dn + \frac{1}{n}\sum_{l=0}^{n-1} cl = dn + \frac{c}{n}\frac{n(n-1)}{2} = (d + \frac{c}{2})n - \frac{c}{2} \ .$$

Therefore, $(d + \frac{c}{2})n - \frac{c}{2} \leq cn$, if $d + \frac{c}{2} \leq c$. Choose $c \geq 2d$.

(b) **Procedure** `medianofmedians`

Input: array a of length n
Output: the median of medians

if $n \leq 5$ **then**
  return `median5`(a);
fi
$m := \lfloor n/5 \rfloor$;
for $i := 1$ **to** m **do**
  $b :=$ `median5`$(a[5i-4, \ldots, 5i])$;
end
return `selection`$(b, \lceil m/2 \rceil)$;

(c) `selection` takes $\Theta(n)$ time in the worse case when `medianofmedians` is used for finding pivots. We sketch the proof below.

Let p be the value returned by `medianofmedians(a)`. Since $b[i]$ is the median of $a[5i-4,\ldots,5i]$, at least three elements in $a[5i-4,\ldots,5i]$ are less than $b[i]$. Moreover, the fact that p is the median of b implies that at least $m/2$ elements in b are less than or equal to p, which means that at least $3m/2$ elements in a are less than or equal to p. However, $m = \lfloor n/5 \rfloor \geq (n-4)/5$, thus at least $(3n-12)/10$ elements in a are less than or equal to p. On the other hand, at most $(7n+12)/10$ elements in a are strictly larger than p.

Let $t(n)$ be the worst case time for `selection`(a, s). Since `partition` and constructing b takes linear time and after a loop there are at most $(7n+12)/10$ to be considered, we have

$$t(n) \leq dn + t(\lfloor n/5 \rfloor) + \max\{t(m) \mid m \leq (7n+12)/10\} ,$$

for some constant d. To prove that $t(n) \in \mathcal{O}(n)$, we need to find a constant c such that $t(n) \leq cn$ for all $n \geq 1$. We proceed by constructive induction on n.

Basis With n_0 to be determined later, we have freedom to choose the constant c such that $t(n) \leq cn$ for all $1 \leq n \leq n_0$, i.e. choose $c \geq t(n)/n$.

Inductive step Consider any integer $n > n_0$. By the induction hypothesis, we have $t(m) \leq cm$ when $1 \leq m < n$. Now we need to find c such that $t(n) \leq cn$ holds.

$$\begin{aligned} t(n) &\leq dn + t(\lfloor n/5 \rfloor) + \max\{t(m) \mid m \leq (7n+12)/10\} \\ &\leq dn + n/5 + (7n+12)/10 \text{ (by the induction hypothesis)} \\ &= cn - (c/10 - d - 6c/5n)n \end{aligned}$$

Therefore, $t(n) \leq cn$ if $(c/10 - d - 6c/5n) \geq 0$, which is equivalent to

$$c \geq \frac{10d}{1 - \frac{12}{n}}$$

when $n \geq 13$. Therefore, any choice of $n_0 \geq 12$ and

$$c \geq \frac{10d}{1 - \frac{12}{n_0+1}}$$

would make the induction step correct. For instance, choose $n_0 = 23$ and $c \geq 20d$. So, it suffices to set

$$c = \max(20d, \max\{t(m)/m \mid 1 \leq m \leq 23\}$$

to conclude that $t(n) \leq cn$ for all $n \geq 1$.

2. See execise sheet 5.