Analysis of Algorithms

Assignment: midterm 2009

Name: .....

Id: .....

Grade

Good Luck!

1. Let $\mathcal{S}$ be a set of $n > 0$ distinct integers. Assume that n is a power of 3. A **ternary comparison** can compare **three** numbers from the set $\mathcal{S}$ and order them from the largest to the smallest.

   (a) Describe an efficient algorithm that uses as few as possible ternary comparisons to find the largest number in the set $\mathcal{S}$. Explain why your algorithm is correct and state the exact number of ternary comparisons it uses in the worst case.

(b) Describe an efficient algorithm that uses as few as possible ternary comparisons to find both the largest and the smallest numbers in the set $\mathcal{S}$. Explain why your algorithm is correct and state the exact number of ternary comparisons it uses in the worst case.

(c) Describe an efficient algorithm that uses as few as possible ternary comparisons to find both the largest and the second largest numbers in the set $\mathcal{S}$. Explain why your algorithm is correct and state the exact number of ternary comparisons it uses in the worst case.

(d) Prove a lower bound on the number of ternary comparisons that are needed to find the largest number in the set $\mathcal{S}$.

(e) Prove a lower bound on the number of ternary comparisons that are needed to find the largest and the smallest numbers in the set $\mathcal{S}$.

2. Let $\mathcal{S}$ be a set of $n > 0$ distinct integers. Assume that n is a power of k for some $k > 0$. A **k-ary comparison** can compare k numbers from the set $\mathcal{S}$ and order them from the largest to the smallest.

   (a) Describe an efficient algorithm that uses as few as possible k-ary comparisons to find the largest number in the set $\mathcal{S}$. Explain why your algorithm is correct and state the exact number of ternary comparisons it uses in the worst case.

(b) Describe an efficient algorithm that uses as few as possible k-ary comparisons to find both the largest and the smallest numbers in the set $\mathcal{S}$. Explain why your algorithm is correct and state the exact number of ternary comparisons it uses in the worst case.

(c) Describe an efficient algorithm that uses as few as possible k-ary comparisons to find both the largest and the second largest numbers in the set $\mathcal{S}$. Explain why your algorithm is correct and state the exact number of ternary comparisons it uses in the worst case.

(d) Prove a lower bound on the number of k-ary comparisons that are needed to find the largest number in the set $\mathcal{S}$.

(e) Prove a lower bound on the number of k-ary comparisons that are needed to find the largest and the smallest numbers in the set $\mathcal{S}$.