CS584
HW#1

DUE: Monday, September 21

1. (4 points) Describe a procedure *P* to convert virtually any sorting algorithm *A* to a new algorithm $P(A)$ with a best-case running time which is linear in the number of elements being sorted and such that the asymptotic average-case and worst-case running times of $P(A)$ are the same as the asymptotic average-case and worst-case running times of *A*.

2. (5 points) Show that any algorithm which finds a member of $A[1..n]$ which is greater than or equal to the median must make at least $\lfloor n/2 \rfloor$ pairwise comparisons, even in the best case. The algorithm can return any $j, 1 \le j \le n$, such that $\left|\{k \mid A[k] \le A[j]\}\right| \ge \lfloor n/2 \rfloor$.

3. (14 points) Given sorted arrays $A[1..n]$ and $B[1..n]$, we want to combined them into a sorted array $C[1..2n]$. That is, if *n*=4 and $A = (3,3,12,410)$ and $B = (-12,3,22,410)$, then $C = (-12,3,3,3,12,22,410,410)$.

a Show that if the lists are combined using pairwise comparisons, then at least $2n-1$ comparisons are necessary in the worst-case.

b Show that if the lists are combined using pairwise comparisons, then $2n-1$ comparisons suffice in the worst-case.

4. (14 points) An *authority* in a digraph $G = (V, E)$ is a vertex of indegree *n*-1 and outdegree 0. That is, *v* is an authority if every other vertex has an edge to *v* and no edge goes out of *v*. Assume that *G* is presented to your algorithm as an adjacency matrix, A_G, where

$$A_G[i, j] = \begin{cases} 1, \text{ if } ij \in E \\ 0, \text{ otherwise} \end{cases}.$$

Construct an algorithm to test if *G* has a an authority. Your algorithm should work in time in $O(n)$. Since A_G has n^2 elements, you should assume that it is already in memory; you are not charged for reading it into memory. (Hint: Consider identifying vertices that are **not** authorities.)

5. (6 points) Assuming that $f, g : \mathbb{N} \to \mathbb{R}^+$, prove or give a counterexample to each of the following.

a $f(n) + g(n) = \Theta\left(\min\left(f(n), g(n)\right)\right)$

b $f(n) = O\left(g(n)\right)$ implies that $2^{f(n)} = O\left(2^{g(n)}\right)$

c $f(n) + o\left(f(n)\right) = \Theta\left(f(n)\right)$

C.S.584
Solutions for H.W. #1

1. $P(A)$: **if** the input array is sorted **then** return the input array $\Theta(n)$
else run algorithm *A*

2. Assume $A[j]$ is chosen to be greater than or equal to the median, and fewer than $\lfloor n/2 \rfloor$ pairwise comparisons have been done. This means that more than $\lfloor n/2 \rfloor$ elements have never been determined to be less than another element of *A*. Change *A* by making every element aside from *j* and the elements which have never been determined to be less than an element of *A* to be ∞. In the new array, every comparison done by the algorithm yields the same outcome. So the algorithm must still identify $A[j]$ as greater than or equal to the median. But since more than $\lfloor n/2 \rfloor$ elements are ∞, this is wrong. So, by contradiction, the algorithm can not exist.

3. ***a*** Assume that there is an algorithm to combine the arrays using fewer than 2*n*-1 pairwise comparisons. Choose *A* and *B* to satisfy

$$A[1] < B[1] < A[2] < B[2] < \ldots < A[n] < B[n]$$

There must be at least one inequality, say $A[i] < B[i]$ or $B[i] < A[i+1]$, above which is not tested by the algorithm. Interchange the two elements of the inequality ($A[i]$ and $B[i]$ or $B[i]$ and $A[i+1]$). The algorithm receives the same answers from the data as before and returns the same answer. But the answer is not correct, and by contradiction the algorithm can not exist.
b Algorithm MERGE of the text combines the two lists using in the worst-case 2*n*-1 pairwise comparisons.

4. We first eliminate *n*-1 vertices which are **not** *cand*idates for being an authority.

$cand \leftarrow 1$
for $i \leftarrow 2$ **to** *n* **do**
  if $(A[cand, i] = 1) \wedge (A[i, cand] = 0)$ **then** $cand \leftarrow i$

No element other than *cand* can be an authority. We now check if *cand* is indeed an authority.

for $i \leftarrow 1$ **to** *n* **do**
  if $(i \neq cand) \wedge ((A[i, cand] = 0) \vee (A[cand, i] = 1))$
    then return "There is no authority"
return "*cand* is an authority"

5. ***a*** false Let $f(n)=1$ and $g(n)=n$. Then $f(n)+g(n)=n+1$ and $\min(f(n)+g(n))=1$ for all $n\geq 1$.

$$f(n)+g(n)=\Theta(\min(f(n),g(n)))\rightarrow n+1=\Theta(1)\rightarrow(\exists c)(\exists n_0)(\forall n\geq n_0)\,n+1\leq c$$

which is clearly false.

b false Let $f(n)=2n$ and $g(n)=n$. It follows that $f(n)=O(g(n))$. But if $2^{f(n)}=O(2^{g(n)})$, then there must exist *c* and n_0 such that for all $n\geq n_0$, $2^{2n}\leq c2^n$. Dividing both sides by 2^n, for all *n* sufficiently large it must be the case that $2^n\leq c$, which is impossible.

c true $f(n)+o(f(n))=\Theta(f(n))$ By the definition of *o*, for any $g(n)$ in $o(f(n))$, $(\forall c>0)(\exists n_0)(\forall n\geq n_0)\,f(n)<cg(n)$. Choosing *c*=1 we get that for any $g(n)$ in $o(f(n))$, $(\exists n_0)(\forall n\geq n_0)\,g(n)<f(n)$ and $f(n)+o(f(n))<2f(n)$. Likewise $f(n)\leq f(n)+o(f(n))$, yielding the claim $f(n)\leq f(n)+o(f(n))<2f(n)$.