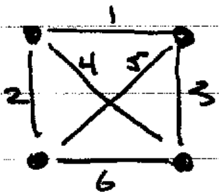
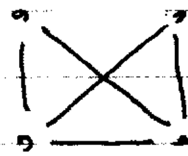
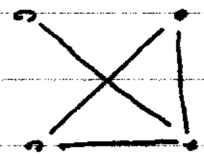
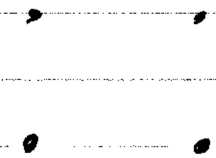
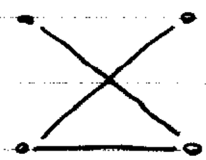
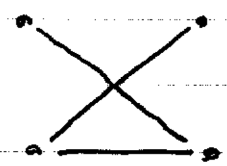
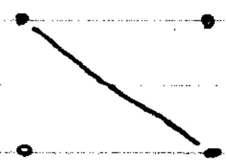
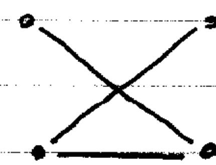
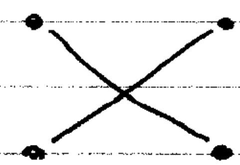
CSE 102 12-2-21

- SETs: Due Sun Dec. 5
- Final Exam: Thur. Dec. 9
- mid2 grading: by Friday Dec. 3
- hw7 tonight:

Illustrate proof of Thm ($\binom{n}{2}$ edge probes necessary to dermine connectedness) in case $n = 4$

<u>Ex.</u> $n=4$ $A = E(K_4)$, $B = \emptyset$

K_4 

edge probe	A	B
1		
2		
3		
4		
5		

$$\left|E(K_4)\right| = \binom{4}{2} = \frac{\cancel{4}\cdot 3\cdot 2}{\cancel{2\cdot 2}} = 6, \text{ so stop.}$$

Exercise

let $b = x_1 x_2 x_3 x_4 x_5 \in \{0,1\}^5$. Problem: determine whether b contains '111'.

allowable operation: peek at a bit.

note: 5 peeks are sufficient

- find an algorithm that uses 4 peeks, express as a decision tree.
- use an adversary argument to show that 4 peeks are necessary.
- do same thing for bit strings of length n, substring "11…1" of length k.

Ex.

Problem: determine (min, max) in an array of length n. Recall we can do this with $\lceil \frac{3n}{2} \rceil - 2$ array comparisons.

lets show $\left(\lceil \frac{3n}{2} \rceil - 2\right)$ comparisons are necessary

Exercise a decision tree argument a lower bound of

$$\lceil \lg (n^2 - n + 1) \rceil = \Omega(\log n)$$

comparisons.

Thm

any algorimth that performs only array comp.s must do at least

$$\left\lceil \frac{3n}{2} \right\rceil - 2$$

such comps to determine both max and min of an array of length n.

Proof.

Run any algorithm for this problem against following adversary.

maintain an array of length n containing marks $+$, $-$

	1	2	3	4		$n-1$	n
A	$\pm$	$\pm$	$\pm$	$\pm$	. . .	$\pm$	$\pm$

each array element has 4 poss. states

$\pm$	candidate for both max min
$+$	" " max only
$-$	" " min only
N (no mark)	neither max nor min.

Each comparison $A[i] < A[j]$ requires adversary to commit one element (loser) as smaller, $\therefore$ not the max, and one element (winner) as larger, $\therefore$ not min.

The adversary's answer depends on current marks on $A[i]$ and $A[j]$

see Page 11 of handout.

As the algorithm performs its queries $A[i] < A[j]$, and marks are removed by adversary, maintain 3 variables

c_0 = # comparisons that remove no marks

c_1 = " " " " 1 mark

c_2 = " " " " 2 mark

Assume algorithm halts and returns a verdict after doing less than $\lceil \frac{3n}{2} \rceil - 2$ comparisons.

<u>Exercise</u>: $\lceil \frac{3n}{2} \rceil - 2 = 2n - 2 - \lfloor \frac{n}{2} \rfloor$

Thus: $c_0 + c_1 + c_2 < 2n - 2 - \lfloor \frac{n}{2} \rfloor$

$$\Rightarrow \quad c_1 + 2c_2 < (2n-2) + (c_2 - \lfloor \tfrac{n}{2} \rfloor) - c_0$$

$$\Rightarrow \quad c_1 + 2c_2 < (2n-2)$$

why last step?

- $c_0 \geq 0$ so $-c_0 \leq 0$

- $c_2 \leq \lfloor \frac{n}{2} \rfloor$ (why: there are at most $\lfloor \frac{n}{2} \rfloor$ pairs of elements from which to remove 2 marks.)

$\therefore$ $c_2 - \lfloor \frac{n}{2} \rfloor \leq 0$

so we have $c_1 + 2c_2 < 2n-2$

The total # of marks removed is therefore $0 \cdot c_0 + 1 \cdot c_1 + 2 \cdot c_2 = c_1 + 2c_2$, and hence

$$\text{\# marks removed} < 2n-2$$

$\therefore$ # marks remaining $> 2n-(2n-2) = 2$

$\therefore$ $\boxed{\text{\# marks remaining} \geq 3}$

so there must still exist either 2 +'s or 2 −'s.

- suppose 2 + signs remain, whatever the algorithm claims as maximum, there is one other candidate that the adversary can use to contradict it.
- If two − signs remain, a similar contradiction arises for min.

∴ No correct algorithm can do less than $\lceil \frac{3n}{2} \rceil - 2$ comparisons.

∴ $\lceil \frac{3n}{2} \rceil - 2$ is a lower bound on # comparisons by any correct algorithm.