

CS200 - Problem Set 5

1. *Party-trick Proof* [11 points] Suppose you are at a party with 19 acquaintances (so there are 20 people at the party). Prove (using a proof by contradiction) that there must be at least two people at the party who talked to the same number of people over the course of the evening. (Note: we assume that if Yan talked to Jan, that also means that Jan talked to Yan.)
2. Fill in the missing parts of the proof that Algorithm 1 for binary search is correct.

Input : (1) Array A containing integers, where there are no repeated integers and the integers are sorted from smallest to largest, (2) an element V in A , and (3) two indices s and f , where $s \leq f$ and the index of V is between s and f (inclusive)

Output: Index g such that $A[g] = V$, and $s \leq g \leq f$.

```
// Base Case
1 if  $f - s = 0$  then
2 | return  $s$ ;
3 end
// Recursive step
4  $mid = \lfloor (f + s)/2 \rfloor$ ;
//  $\lfloor \cdot \rfloor$  means round down to the nearest integer
5 if  $A[mid] = V$  then
6 | return  $mid$ ;
7 else
8 | if  $A[mid] < V$  then
9 | | return BinarySearch( $A, V, mid + 1, f$ );
10 | else
11 | | return BinarySearch( $A, V, s, mid - 1$ );
12 | end
13 end
```

Algorithm 1: BinarySearch(A, V, s, f)

- (a) [3 points] Fill in the blanks in the introduction:

Let $P(n)$ be the predicate that BinarySearch(A, V, s, f) finds the position of V in A , where $n = \underline{\hspace{2cm}}$. We will prove using strong induction that $P(n)$ is true for all $n \geq \underline{\hspace{2cm}}$.

- (b) [3 points] Fill in the blank in the base case:

When $n = \underline{\hspace{2cm}}$, we enter the base case of the algorithm, and since $f = s$, we know the index of V must be $f = s$, since the index of V must be between s and f inclusive. We see in this case, the algorithm returns s , so it is correct.

(c) **[3 points]** Fill in the blanks in the inductive step:

Inductive step: Let $k \geq$ _____. We assume for strong induction that `BinarySearch` is correct whenever _____ = j , for all j such that $k \geq j \geq$ _____. Now we consider the case that _____ = $k+1$. Since $k+1 >$ _____, we go to the recursive step of the algorithm. There are three cases to consider, corresponding to the if-else conditionals: $A[mid] =$ _____, $A[mid] <$ _____ and $A[mid] >$ _____.

(d) **[6 points]** Fill in the argument in the second case:

In the first case, $A[mid] = V$. Then we return mid , the index of V , which is correct. In the second case, $A[mid] < V$. Let g be the index such that $A[g] = V$. Then

(e) **[3 points]** Fill in the final blanks:

In the third case, _____. Using a nearly argument as in the second case, we have that the algorithm is correct in this case. Thus, in all three cases, $P(k+1)$ is true.

Therefore, $P(n)$ is true $\forall n \in \mathbb{Z} : n \geq$ _____.

3. **[3 points]** Explain why we needed to use strong induction instead of regular induction in the proof of binary search. As an example, talk about what happens in the algorithm if the input array is size 5.