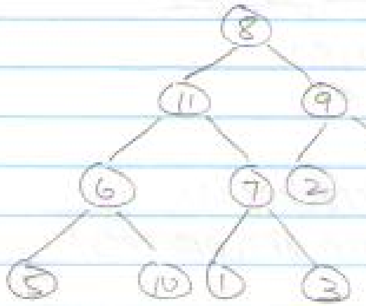
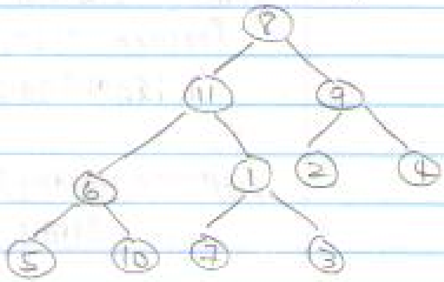
Eleanor Wong
32507121 k1w8
Lawrence Garcia
4101828 j3x8

CPSC 221 - Written Assignment 2

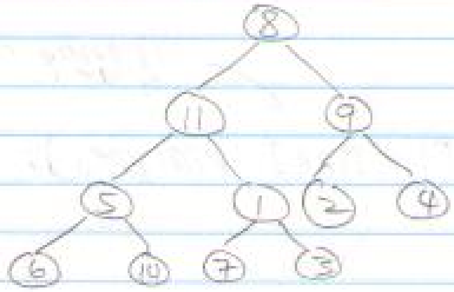
1.

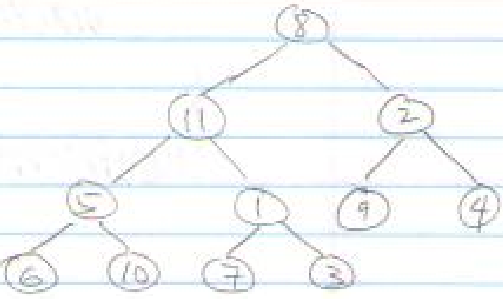




8, 11, 9, 6, 7, 2, 4, 5, 10, 1, 3

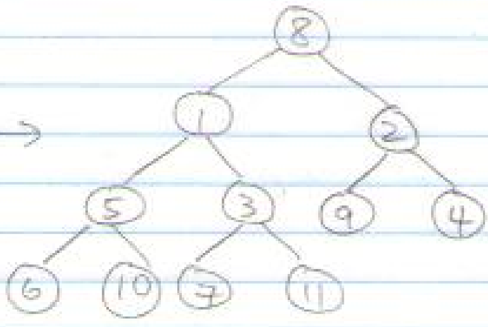
8, 11, 9, 6, 1, 2, 4, 5, 10, 7, 3

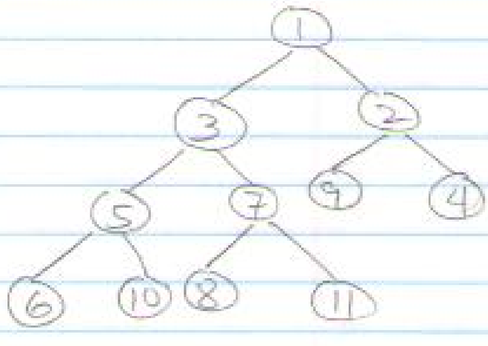




8, 11, 9, 5, 1, 2, 4, 6, 10, 7, 3

8, 11, 2, 5, 1, 9, 4, 6, 10, 7, 3





8, 1, 2, 5, 3, 9, 4, 6, 10, 7, 10

1, 3, 2, 5, 7, 9, 4, 6, 10, 8, 11

Eleanor Wong
32507121 k1w8

Lawrence Garcia
4101828 j3x8

```
2) function countUnmatchedBrackets (~~strin character~~ charArray):
     return countBrackets (charArray, 0);
     if (charArray == NULL) return 0;

   function countBrackets ( array, brackets):
       if ( array. isEmpty()) return abs(brackets);

       if (array. getFirst() == "(") {
           brackets ++;
       }
       if (array. getFirst() == ")") {
           brackets --;
       }
                              sub array removing first
       countBrackets (array[1 → size], brackets);
   }
```

Eleanor Wong
32507121 k1w8

Lawrence Garcia
4101828 j3x8

3.
```
int bitsort( int *A, int n) {
    int i=0;
    int j=n-1;
    while ( i < j ) {
        if ( A[i] == 0 ) {
            i++;
        } else if ( A[j] == 1 ) {
            j--;
        } else {
            swap (A[i], A[j]);
        }
    }
    return j;
}
```

(a) Relevant code reproduced and filled in above.

(b) Loop invariant:

All entries to the left of index i are 0's. and all entries to the right of index j are 1's.

Base case (before the loop is entered):

$i=0$ and $j=n-1$ so there are no entries to the left of i and to the right of j so the loop invariant holds.

Inductive step:

- Case 1: $A[i] == 0$

  All entries to the left of i are 0. Incrementing i will put $A[i]$ to the left of the new i. This element is also 0 so the loop invariant holds.

- Case 2: $A[j] == 1$

  By the loop invariant before entering the loop, all entries to the right of j are 1. Decrementing j will put

Eleanor Wong
32507121 k1w8
Lawrence Garcia
4101828 j3x8

the current $A[j]$, which is 1, to the left of the new j index, so the loop invariant holds.

- Case 3: $A[i] == 1$ and $A[j] == 0$

Swapping the elements does not alter the loop invariant because the elements to the left of i and right of j are untouched and neither i nor j are changed. But swapping ensures that the next iteration is not Case 3.

Loop termination: The condition is $i < j$ which will eventually terminate because in Case 1, i is incremented, in Case 2, j is decremented and Case 3 swaps to ensure that the next iteration is Case 1. This ensures that eventually, after a certain number of iterations, $i == j$ and the loop terminates.

(c) When the loop ends, $i == j$ because of the initial values assigned to each. By the loop invariant, every element to the left of i is 0 and every element to the right of j is 1. The element at index i (which is now equal to j) can be either 1 or 0, but this doesn't violate the sorting. So all the zeros come before the ones.

index	0	1	2	...				N-1
	0	0	0	0	0 or 1	1	1	1

← at the end of the loop

$i^{th} == j^{th}$ index

Eleanor Wong
32507121 k1w8

Lawrence Garcia
4101828 j3x8

4. (a) Consider the necklace with $n \geq 1$ links and n not divisible by 3.
This means that n can be written as
$n = 3k + r$ where k is an integer, $k \geq 0$
and r is an integer $0 < r \leq 2$.

The loop invariant that Alice must preserve in order to win is that the number of remaining links (after her turn) must be divisible by 3.

Prove this by induction on n

- Consider the base case: $k = 0$ $r = 1$ or 2.
Since Alice goes first, she takes r links and wins.

- Inductive step: Assume that Alice can win if $n = 3k + r$.
Now we want to show that she can win if $n = 3(k+1) + r$

Rewrite the above as $n = \underbrace{(3k + r)}_{1} + \underbrace{3}_{2}$

By our assumption, Alice will have removed the last link of the $3k + r$ links. That leaves 3 links, of which Bob can remove 1 or 2. Clearly, Alice can take $3 - b$ links during her turn, where b is the number of links Bob removes. Thus the $n = 3(k+1) + r$ case holds given that the $n = 3k + r$ case holds.

This completes the proof by induction.

Alternatively, we can also use a loop invariant to show this.

Eleanor Wong
32507121 k1w8

Lawrence Garcia
4101828 j3x8

Again, consider $n = 3k + r$ starting links with $k \geq 0$ and k an integer, and $r = 1$ or 2.

Consider the loop as starting with Bob's first turn, after Alice's first turn.

During her first turn, Alice removes r links, which leaves $n = 3k$ links. So the loop invariant holds before we enter the loop.

- Now we enter the loop, with each "iteration" a set of turns from each player. If Bob removes b links, Alice must remove $3-b$ links to preserve the loop invariant.
- while (n > 0) {

      Bob removes b links
      Alice removes 3 - b links
      // 3 total links are removed in each iteration of the loop.
      // so the loop invariant holds at the end of each loop.
  }

The loop must terminate when $n = 0$, which will happen after k iterations. Since Alice is the last one to remove links to make n equal to zero, Alice always wins.

Eleanor Wong
32507121 k1w8

Lawrence Garcia
4101828 j3x8

$16{,}777{,}216 = 256^3$
$65536 = 256^2$
65535

5) $m = 2^{16} - 1$

$h(s_0 s_1 \cdots s_k) = (s_0 + s_1 \cdot 256 + s_2 \cdot 256^2 + \dots + s_k \cdot 256^k)$ ← each s_i is an 8 bit character

For three character strings, $\boxed{k=2}$

$$h(s_0 s_1 s_2) = (s_0 + s_1 \cdot 256 + s_2 \cdot 256^2) \bmod m$$

For each character s_i, there are $2^8 = 256$ possible characters

Three character strings $\Rightarrow$ 256^3 possible combination

so given $x \in [0, 255]$

$x \bmod m = x$

$(x + 1 * m) \bmod m = x$

$\vdots$

$(x + 255 * m) \bmod m = x$

But $(x + 256 * m) \bmod m = x$

$\Rightarrow$ $2^8 (2^{16} - 1) \bmod (2^{16} - 1) = 0$

(because $(c \cdot m) \bmod m = 0$ $c \in \mathbb{Z}$

so $\exists$ at least one slot st.

$x \in [0, 255]$ +1 hash to the same slot.

b) Let $h(s_0 s_1 \cdots s_r) = (s_0 + s_1 2^8 + s_2 (2^8)^2 \cdots) \bmod m$

suppose x_i is a character of string x of length n

y is a permutation of x. Since cycles can be decomposed into two cycles then

$x_a = y_b$ and $y_a = x_b$ as the permutation difference between x, y

Assume $a > b$

$[h(x) - h(y)] \bmod m = [(x_a) 2^{8a} + (x_b) 2^{8b} - y_a 2^{8a} - y_b 2^{8b}] \bmod m$

$y_a = x_b$, $x_a = y_b$: $[x_a 2^{8a} - x_a 2^{8b} + x_b 2^{8b} - x_a 2^{8a}) \bmod m$

$= [x_a [2^{8(a-b)}] + x_b [2^{8(b-1)}]] \bmod m$

$= [x_a - x_b][2^{8a} - 2^{8b}] \bmod m$

For the case $a = 2$, $b = 0$,

$[h(x) - h(y)] \bmod m = 0$

$\Rightarrow$ $\exists$ permutations of x, y st. $h(x) = h(y)$.

More generally, if characters at positions a, b st.

$[2^{8a} - 2^{8b}] \bmod m = 0$, they will hash to the same spot.