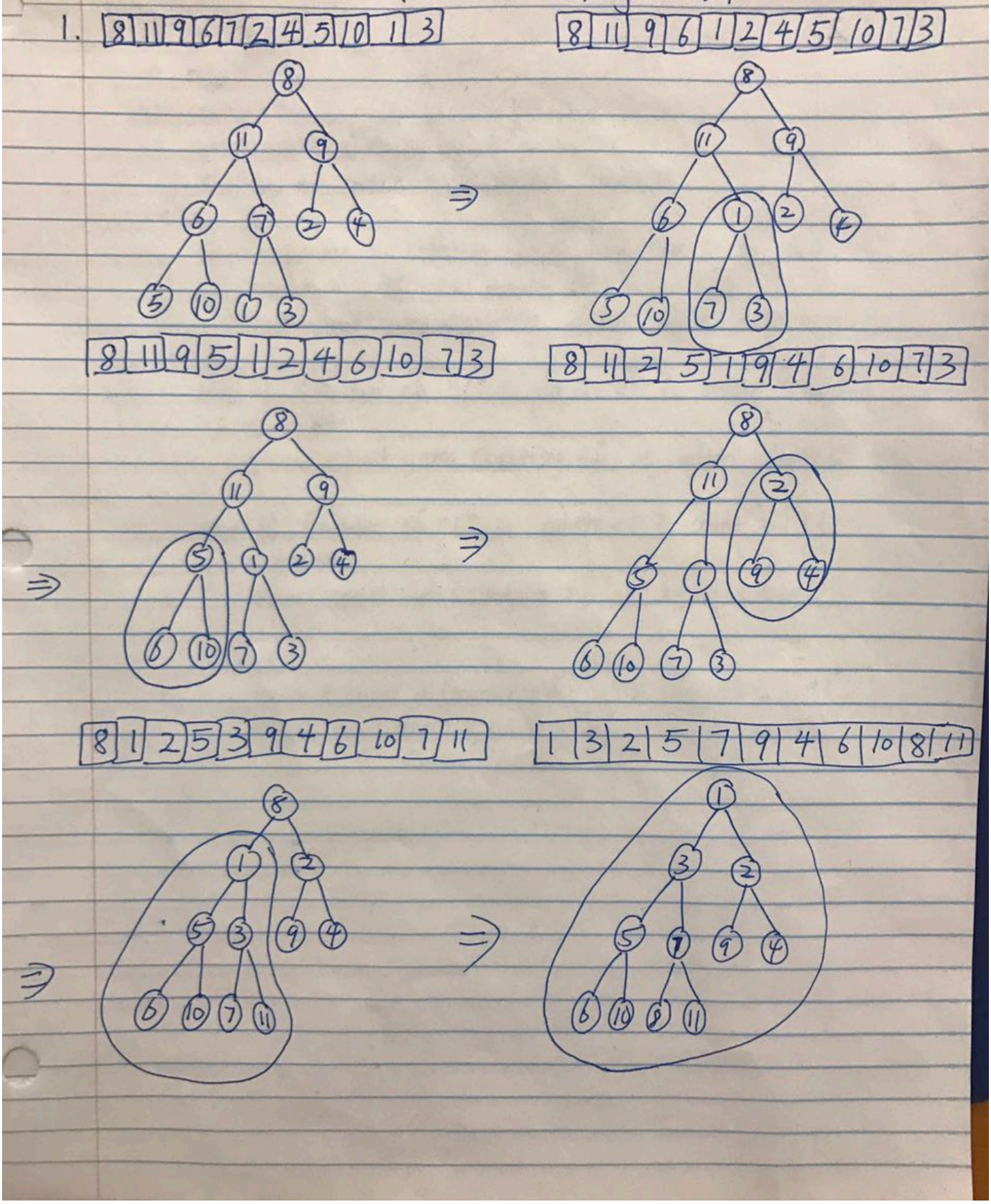
Zongming Liu(j4v9a) Yu Guo(b4m0b)

CPSC 221 Assignment 2

1.



Zongming Liu(j4v9a) Yu Guo(b4m0b)

2.

```
int result=0;
int count=0;
int count_unma (string s) {
  if (s does not contain ")") return result;
  else {
    string substr = substring of s from char 0 to first ")";
    if (number of "(" in substr == 1) {
      return count_unma (substring of s after substr);
    }
    else if (number of "(" in substr <1 & count <1) {
    ++ result;
      return count_unma (substring of s after substr);
    }
    else if (number of "(" in substr<1 & count >=1) {
    -- count;
      return count_unma (substring of s after substr);
    }
    else {
      count = count + (number of "(" in substr) -1;
      return count_unma (substring of s after substr);
    }
  }
}
```

Zongming Liu(j4v9a) Yu Guo(b4m0b)

3.

(a):

```
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;
}
```

(b): Loop invariant: At the beginning of the iteration, the left i are not 1 and the right of j are not 0.

Therefore, at each iteration, if A[i] == 0, then i++ still preserves the invariant; else if A[j] == 1, --j also preserves the invariant; else if A[i] == 1 & A[j] == 0, swap A[i] & A[j] still preserves the invariant.

(c): Before the beginning of the iteration, the left of i are not 1's and the right of j are not 0's, and this still holds after each iteration. The loop terminates when i = j, which makes the list sorted since the left of i are all 0's and right of j's are all 1's.

(d): When the loop terminates, A[j] = 1.

Zongming Liu(j4v9a) Yu Guo(b4m0b)

4.

(a): [redacted]

Proof: Base case: $n=1$, then Alice will win.
$n=2$, then Alice will win.

Inductive hyp.

Inductive ~~Step~~: ① Assume $n=k$, that $k \bmod 3 = 1$, and Alice will win. Alice will take one first, then the left will be divisible by 3, in which Bob can not win, cause either he takes one or two, will always make Alice be able to get the rest.

Inductive step: let $n=k+1$, ~~that k=n~~ then ~~k=n~~ $n \bmod 3 = (k+1) \bmod 3 = 2$. Then Alice will take 2 first, to keep the rest be divisible by 3. Bob cannot win regardless one or two he will take.

② Inductive hyp.: Assume $n=k$, that $k \bmod 3 = 2$, and Alice will win.

Inductive Step: let $n=k+2$, so ~~A=k+2~~ $n \bmod 3 = (k+2) \bmod 3 = 1$

Thus, Alice will take one first, to ~~a~~ keep the rest be divisible by 3. ~~Bob~~ Either Bob takes one or two will always leave ~~one~~ two or one for Alice to pick, Alice will always win.

Q.E.D.

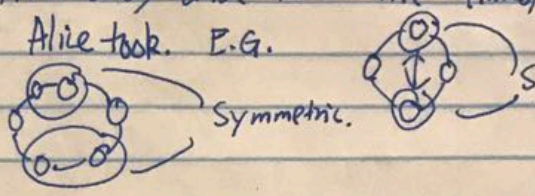
Invariant: Alice will always keep the remaining links be divisible by 3.

Zongming Liu(j4v9a) Yu Guo(b4m0b)

(b): When $n=3$, Bob will win: Alice takes one, Bob takes 2; Alice takes 2, Bob takes 1.

When $n \geq 3$

if n is even: Then Bob can just take the same number of Alice did, and take the links that is symmetric to Alice took. E.G.



if n is odd: Then Bob will take one if Alice first takes two, and Bob will take two if Alice first takes one; also, Bob will take the link to divide the remaining links to be two equal number of links. Then, the remaining n would be even, Bob can take symmetric stated above.

5.

(a): Since there are $(256+256^2+256^3) = 16843008$ possible three characters strings, and there are $2^{16}-1 = 65535$ total slots, thus, if we are going to insert all 16843008 into 65535 slots, there is going to be a slot with at least $\frac{16843008}{65535} \approx 257$ elements (strings).

(b): When the permutation works to make the distances of every character's V position in the new string to its original position in the original string are all even numbers, then those two strings will be hashed into the same slot. Since the permutation will make the difference of those two strings to be divisible by $2^{16}-1 = 65535$.