Homework 1 Answers

Part (a)

```
void f1(int n)
{
    int i=2;
    while(i < n){
        /* O(1) */
        i = i*i;
    }
}
```

Analysis:

- Sequence: $i = 2, 4, 16, 256, \ldots$
- $i_k = 2^{2^k}$
- Find k where $2^{2^k} \geq n \Rightarrow k = \Theta(\log \log n)$

Running Time: $\Theta(\log \log n)$

Part (b)

```
void f2(int n)
{
    for(int i=1; i <= n; i++){
        if( (i % (int)sqrt(n)) == 0){
            for(int k=0; k < pow(i,3); k++) {
              /* O(1) */
            }
        }
    }
}
```

Analysis:

- Outer loop: n
- Condition true $\sqrt{n}$ times
- Inner loop: i^3, max n^3
- Total: $\sum_{m=1}^{\sqrt{n}} (m\sqrt{n})^3 = \Theta(n^{3.5})$

Running Time: $\Theta(n^{3.5})$

Part (c)

```
for(int i=1; i <= n; i++){
  for(int k=1; k <= n; k++){
    if( A[k] == i){
      for(int m=1; m <= n; m=m+m){
        // O(1)
      }
    }
  }
}
```

Analysis:

- Outer loops: n^2
- Condition true once per i
- Innermost loop: $\Theta(\log n)$
- Total: $\Theta(n^2 \log n)$

Running Time: $\Theta(n^2 \log n)$

Part (d)

```
int f (int n)
{
  int *a = new int [10];
  int size = 10;
  for (int i = 0; i < n; i ++)
     {
        if (i == size)
          {
             int newsize = 3*size/2;
             int *b = new int [newsize];
             for (int j = 0; j < size; j ++) b[j] = a[j];
             delete [] a;
             a = b;
             size = newsize;
          }
        a[i] = i*i;
     }
}
```

Analysis:

- Loop: n iterations
- Resizing copies: Geometric series $\sum \frac{3}{2}^k = O(n)$
- Total: $\Theta(n)$

Running Time: $\Theta(n)$

1 Problem 2: Recursive Linked List Manipulation

1.1 C++ Code

```
struct Node {
    int val;
    Node*  next;
};

Node* llrec(Node* in1, Node* in2)
{
    if(in1 == nullptr) {
        return in2;
    }
    else if(in2 == nullptr) {
        return in1;
    }
    else {
        in1->next = llrec(in2, in1->next);
        return in1;
    }
}
```

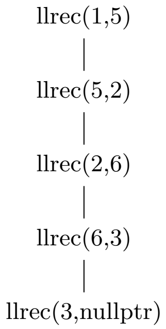
1.2 Question a

What linked list is returned if `llrec` is called with the input linked lists `in1 = 1,2,3,4` and `in2 = 5,6`?

1.2.1 Answer

The returned linked list is: `1 -> 5 -> 2 -> 6 -> 3 -> 4`.

1.2.2 Explanation



a) `llrec(1,5)`:

- `in1` is 1, `in2` is 5.
- `1->next` is set to `llrec(5,2)`.

b) `llrec(5,2)`:

    - `in1` is 5, `in2` is 2.
    - `5->next` is set to `llrec(2,6)`.

c) `llrec(2,6)`:
    - `in1` is 2, `in2` is 6.
    - `2->next` is set to `llrec(6,3)`.

d) `llrec(6,3)`:
    - `in1` is 6, `in2` is 3.
    - `6->next` is set to `llrec(3,nullptr)`.

e) `llrec(3,nullptr)`:
    - `in1` is 3, `in2` is `nullptr`.
    - Returns `in1`, which is 3.

As the recursion unwinds:

- `6->next` is set to `3`.
- `2->next` is set to `6 -> 3`.
- `5->next` is set to `2 -> 6 -> 3`.
- `1->next` is set to `5 -> 2 -> 6 -> 3`.

The final linked list is:

$$1 \to 5 \to 2 \to 6 \to 3 \to 4$$

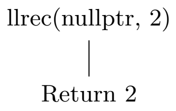
1.3 Question b

What linked list is returned if `llrec` is called with the input linked lists `in1 = nullptr` and `in2 = 2`?

1.3.1 Answer

The returned linked list is: `2`.

1.3.2 Explanation



a) `llrec(nullptr, 2)`:
    - `in1` is `nullptr`, `in2` is 2.
    - Since `in1` is `nullptr`, the function returns `in2`, which is 2.

The final linked list is simply:

$$2$$