Omar Alzibdeh || 1945261
COSC 3320 Algo
Assignment 1, 7 questions
Leiss

Question 1.

To determine the time complexity of the Tower of Hanoi problem, it is imperative to establish a precise algorithmic framework. The Tower of Hanoi problem entails the movement of disks from one peg to another, one step at a time. For instance, if we denote 'n' as the number of disks, let us consider a scenario with three disks for the sake of illustration. To transfer the bottom disk from the Tower of Hanoi to its destination necessitates moving the top disk from peg A1 to peg A3. This entails precise (n-1) movements. Subsequently, similar operations are performed for the remaining disks, each involving (n-1) movements. Therefore, the total minimum number of movements required is determined to be 3(n-1).

Formally expressed, this can be denoted as:

$$f(n) = 3^n-1.$$

Furthermore, each peg must undergo 2n movements, as it must traverse between pegs in both directions, contingent upon the number of pegs. Thus, we arrive at the expression:

$$f(n) = 3^n + 2n - 1.$$

With this equation, we can easily find the time complexity Big O notation of:

$$O(3^n)$$

We then have to find the Big O notation for the space complexity, however, this is much more simple. Each disk will move at a constant right in a binary manner (left or right), without any other complex movements, leaving us with:
O(n)

This analysis establishes a formal representation of the Tower of Hanoi problem's time complexity, considering the movement of disks in a sequential manner.

```
#include <iostream>
#include <stack>
#include <cmath>
using namespace std;

class peg
{
public:
    stack<int> diskStack;
    string pegLabel;

    peg(string pegLabel) : pegLabel(pegLabel) {}

    bool isEmpty()
    {
        return diskStack.empty();
    }
};

int totalMoves = 0;

void moveDisk(peg *fromPeg, peg *toPeg)
{
    if (fromPeg->isEmpty())
    {
        return;
    }
    int disk = fromPeg->diskStack.top();
    fromPeg->diskStack.pop();
    toPeg->diskStack.push(disk);
    totalMoves++;
    cout << totalMoves << ": Move disk " << disk << " from " <<
fromPeg->pegLabel << " to " << toPeg->pegLabel << endl;
}

void TOH(int n, peg *source, peg *dest,
           peg *aux1, peg *aux2,
           peg *aux3, peg *aux4)
{
    if (n == 0)
        return;

    // Move smaller disks from Start to Aux4
    TOH(n - 1, source, aux4, aux1, aux2, aux3, dest);

    moveDisk(source, aux1);
    moveDisk(aux1, aux2);
    moveDisk(aux2, aux3);

    if (n > 2)
    {
        moveDisk(aux3, aux4);
        moveDisk(source, aux1);
        moveDisk(aux1, aux2);
        moveDisk(aux2, aux3);
    }
    moveDisk(aux3, dest);
```

```
    // Now move the n-1 disk to dest using the n-2 pegs available
    TOH(n - 1, aux4, dest, source, aux1, aux2, aux3);
}


int main()
{
    peg start("Start"), aux1("Aux1"), aux2("Aux2"), aux3("Aux3"),
aux4("Aux4"), dest("Dest");

    int N = 1;
    cout << "N = " << N << endl;
    for (int i = N; i > 0; i--)
    {
        start.diskStack.push(i);
    }

    TOH(N, &start, &dest, &aux1, &aux2, &aux3, &aux4);

    return 0;
}
```

OUTPUTS:

n=1 1:Move disk 1 from Start to Aux1 2:Move disk 1 from Aux1 to Aux2 3:Move disk 1 from Aux2 to Aux3 4:Move disk 1 from Aux3 to Dest	n=2 1:Move disk 1 from Start to Aux1 2:Move disk 1 from Aux1 to Aux2 3:Move disk 1 from Aux2 to Aux3 4:Move disk 1 from Aux3 to Aux4 5:Move disk 2 from Start to Aux1 6:Move disk 2 from Aux1 to Aux2 7:Move disk 2 from Aux2 to Aux3 8:Move disk 2 from Aux3 to Dest 9:Move disk 1 from Aux4 to Aux1 10:Move disk 1 from Aux1 to Aux2 11:Move disk 1 from Aux2 to Aux3 12:Move disk 1 from Aux3 to Dest	n=3 1:Move disk 1 from Start to Aux1 2:Move disk 1 from Aux1 to Aux2 3:Move disk 1 from Aux2 to Aux3 4:Move disk 1 from Aux3 to Aux4 5:Move disk 2 from Start to Aux1 6:Move disk 2 from Aux1 to Aux2 7:Move disk 2 from Aux2 to Aux3 8:Move disk 1 from Aux4 to Aux1 9:Move disk 1 from Aux1 to Aux2 10:Move disk 2 from Aux3 to Aux4 11:Move disk 1 from Aux2 to Aux3 12:Move disk 1 from Aux3 to Aux4 13:Move disk 3 from Start to Aux1 14:Move disk 3 from Aux1 to Aux2 15:Move disk 3 from Aux2 to Aux3 16:Move disk 3 from Aux3	n=4 1:Move disk 1 from Start to Aux1 2:Move disk 1 from Aux1 to Aux2 3:Move disk 1 from Aux2 to Aux3 4:Move disk 1 from Aux3 to Aux4 5:Move disk 2 from Start to Aux1 6:Move disk 2 from Aux1 to Aux2 7:Move disk 2 from Aux2 to Aux3 8:Move disk 1 from Aux4 to Aux1 9:Move disk 1 from Aux1 to Aux2 10:Move disk 2 from Aux3 to Aux4 11:Move disk 1 from Aux2 to Aux3 12:Move disk 1 from Aux3 to Aux4 13:Move disk 3 from Start to Aux1 14:Move disk 3 from Aux1 to Aux2 15:Move disk 3 from Aux2 to Aux3 16:Move disk 1 from Aux4

		to Dest 17:Move disk 1 from Aux4 to Aux1 18:Move disk 1 from Aux1 to Aux2 19:Move disk 2 from Aux4 to Aux1 20:Move disk 1 from Aux2 to Aux3 21:Move disk 1 from Aux3 to Aux4 22:Move disk 2 from Aux1 to Aux2 23:Move disk 2 from Aux2 to Aux3 24:Move disk 2 from Aux3 to Dest 25:Move disk 1 from Aux4 to Aux1 26:Move disk 1 from Aux1 to Aux2 27:Move disk 1 from Aux2 to Aux3 28:Move disk 1 from Aux3 to Dest	to Aux1 17:Move disk 1 from Aux1 to Aux2 18:Move disk 2 from Aux4 to Aux1 19:Move disk 1 from Aux2 to Aux3 20:Move disk 1 from Aux3 to Aux4 21:Move disk 2 from Aux1 to Aux2 22:Move disk 1 from Aux4 to Aux1 23:Move disk 1 from Aux1 to Aux2 24:Move disk 3 from Aux3 to Aux4 25:Move disk 1 from Aux2 to Aux3 26:Move disk 1 from Aux3 to Aux4 27:Move disk 2 from Aux2 to Aux3 28:Move disk 1 from Aux4 to Aux1 29:Move disk 1 from Aux1 to Aux2 30:Move disk 2 from Aux3 to Aux4 31:Move disk 1 from Aux2 to Aux3 32:Move disk 1 from Aux3 to Aux4 33:Move disk 4 from Start to Aux1 34:Move disk 4 from Aux1 to Aux2 35:Move disk 4 from Aux2 to Aux3 36:Move disk 4 from Aux3 to Dest 37:Move disk 1 from Aux4 to Aux1 38:Move disk 1 from Aux1 to Aux2 39:Move disk 2 from Aux4 to Aux1 40:Move disk 1 from Aux2 to Aux3 41:Move disk 1 from Aux3 to Aux4 42:Move disk 2 from Aux1 to Aux2 43:Move disk 1 from Aux4 to Aux1 44:Move disk 1 from Aux1 to Aux2 45:Move disk 3 from Aux4 to Aux1 46:Move disk 1 from Aux2 to Aux3

			47:Move disk 1 from Aux3 to Aux4 48:Move disk 2 from Aux2 to Aux3 49:Move disk 1 from Aux4 to Aux1 50:Move disk 1 from Aux1 to Aux2 51:Move disk 2 from Aux3 to Aux4 52:Move disk 1 from Aux2 to Aux3 53:Move disk 1 from Aux3 to Aux4 54:Move disk 3 from Aux1 to Aux2 55:Move disk 3 from Aux2 to Aux3 56:Move disk 3 from Aux3 to Dest 57:Move disk 1 from Aux4 to Aux1 58:Move disk 1 from Aux1 to Aux2 59:Move disk 2 from Aux4 to Aux1 60:Move disk 1 from Aux2 to Aux3 61:Move disk 1 from Aux3 to Aux4 62:Move disk 2 from Aux1 to Aux2 63:Move disk 2 from Aux2 to Aux3 64:Move disk 2 from Aux3 to Dest 65:Move disk 1 from Aux4 to Aux1 66:Move disk 1 from Aux1 to Aux2 67:Move disk 1 from Aux2 to Aux3 68:Move disk 1 from Aux3 to Dest
n=5 1:Move disk 1 from Start to Aux1 2:Move disk 1 from Aux1 to Aux2 3:Move disk 1 from Aux2 to Aux3 4:Move disk 1 from Aux3 to Aux4 5:Move disk 2 from Start to Aux1 6:Move disk 2 from Aux1 to Aux2 7:Move disk 2 from Aux2	n=6 1:Move disk 1 from Start to Aux1 2:Move disk 1 from Aux1 to Aux2 3:Move disk 1 from Aux2 to Aux3 4:Move disk 1 from Aux3 to Aux4 5:Move disk 2 from Start to Aux1 6:Move disk 2 from Aux1 to Aux2 7:Move disk 2 from Aux2	n=7 1:Move disk 1 from Start to Aux1 2:Move disk 1 from Aux1 to Aux2 3:Move disk 1 from Aux2 to Aux3 4:Move disk 1 from Aux3 to Aux4 5:Move disk 2 from Start to Aux1 6:Move disk 2 from Aux1 to Aux2 7:Move disk 2 from Aux2	n=8 1:Move disk 1 from Start to Aux1 2:Move disk 1 from Aux1 to Aux2 3:Move disk 1 from Aux2 to Aux3 4:Move disk 1 from Aux3 to Aux4 5:Move disk 2 from Start to Aux1 6:Move disk 2 from Aux1 to Aux2 7:Move disk 2 from Aux2

to Aux3 8:Move disk 1 from Aux4 to Aux1 9:Move disk 1 from Aux1 to Aux2 10:Move disk 2 from Aux3 to Aux4 11:Move disk 1 from Aux2 to Aux3 12:Move disk 1 from Aux3 to Aux4 13:Move disk 3 from Start to Aux1 14:Move disk 3 from Aux1 to Aux2 15:Move disk 3 from Aux2 to Aux3 16:Move disk 1 from Aux4 to Aux1 17:Move disk 1 from Aux1 to Aux2 18:Move disk 2 from Aux4 to Aux1 19:Move disk 1 from Aux2 to Aux3 20:Move disk 1 from Aux3 to Aux4 21:Move disk 2 from Aux1 to Aux2 22:Move disk 1 from Aux4 to Aux1 23:Move disk 1 from Aux1 to Aux2 24:Move disk 3 from Aux3 to Aux4 25:Move disk 1 from Aux2 to Aux3 26:Move disk 1 from Aux3 to Aux4 27:Move disk 2 from Aux2 to Aux3 28:Move disk 1 from Aux4 to Aux1 29:Move disk 1 from Aux1 to Aux2 30:Move disk 2 from Aux3 to Aux4 31:Move disk 1 from Aux2 to Aux3 32:Move disk 1 from Aux3 to Aux4 33:Move disk 4 from Start to Aux1 34:Move disk 4 from Aux1 to Aux2 35:Move disk 4 from Aux2 to Aux3 36:Move disk 1 from Aux4 to Aux1 37:Move disk 1 from Aux1 to Aux2	to Aux3 8:Move disk 1 from Aux4 to Aux1 9:Move disk 1 from Aux1 to Aux2 10:Move disk 2 from Aux3 to Aux4 11:Move disk 1 from Aux2 to Aux3 12:Move disk 1 from Aux3 to Aux4 13:Move disk 3 from Start to Aux1 14:Move disk 3 from Aux1 to Aux2 15:Move disk 3 from Aux2 to Aux3 16:Move disk 1 from Aux4 to Aux1 17:Move disk 1 from Aux1 to Aux2 18:Move disk 2 from Aux4 to Aux1 19:Move disk 1 from Aux2 to Aux3 20:Move disk 1 from Aux3 to Aux4 21:Move disk 2 from Aux1 to Aux2 22:Move disk 1 from Aux4 to Aux1 23:Move disk 1 from Aux1 to Aux2 24:Move disk 3 from Aux3 to Aux4 25:Move disk 1 from Aux2 to Aux3 26:Move disk 1 from Aux3 to Aux4 27:Move disk 2 from Aux2 to Aux3 28:Move disk 1 from Aux4 to Aux1 29:Move disk 1 from Aux1 to Aux2 30:Move disk 2 from Aux3 to Aux4 31:Move disk 1 from Aux2 to Aux3 32:Move disk 1 from Aux3 to Aux4 33:Move disk 4 from Start to Aux1 34:Move disk 4 from Aux1 to Aux2 35:Move disk 4 from Aux2 to Aux3 36:Move disk 1 from Aux4 to Aux1 37:Move disk 1 from Aux1 to Aux2	to Aux3 8:Move disk 1 from Aux4 to Aux1 9:Move disk 1 from Aux1 to Aux2 10:Move disk 2 from Aux3 to Aux4 11:Move disk 1 from Aux2 to Aux3 12:Move disk 1 from Aux3 to Aux4 13:Move disk 3 from Start to Aux1 14:Move disk 3 from Aux1 to Aux2 15:Move disk 3 from Aux2 to Aux3 16:Move disk 1 from Aux4 to Aux1 17:Move disk 1 from Aux1 to Aux2 18:Move disk 2 from Aux4 to Aux1 19:Move disk 1 from Aux2 to Aux3 20:Move disk 1 from Aux3 to Aux4 21:Move disk 2 from Aux1 to Aux2 22:Move disk 1 from Aux4 to Aux1 23:Move disk 1 from Aux1 to Aux2 24:Move disk 3 from Aux3 to Aux4 25:Move disk 1 from Aux2 to Aux3 26:Move disk 1 from Aux3 to Aux4 27:Move disk 2 from Aux2 to Aux3 28:Move disk 1 from Aux4 to Aux1 29:Move disk 1 from Aux1 to Aux2 30:Move disk 2 from Aux3 to Aux4 31:Move disk 1 from Aux2 to Aux3 32:Move disk 1 from Aux3 to Aux4 33:Move disk 4 from Start to Aux1 34:Move disk 4 from Aux1 to Aux2 35:Move disk 4 from Aux2 to Aux3 36:Move disk 1 from Aux4 to Aux1 37:Move disk 1 from Aux1 to Aux2	to Aux3 8:Move disk 1 from Aux4 to Aux1 9:Move disk 1 from Aux1 to Aux2 10:Move disk 2 from Aux3 to Aux4 11:Move disk 1 from Aux2 to Aux3 12:Move disk 1 from Aux3 to Aux4 13:Move disk 3 from Start to Aux1 14:Move disk 3 from Aux1 to Aux2 15:Move disk 3 from Aux2 to Aux3 16:Move disk 1 from Aux4 to Aux1 17:Move disk 1 from Aux1 to Aux2 18:Move disk 2 from Aux4 to Aux1 19:Move disk 1 from Aux2 to Aux3 20:Move disk 1 from Aux3 to Aux4 21:Move disk 2 from Aux1 to Aux2 22:Move disk 1 from Aux4 to Aux1 23:Move disk 1 from Aux1 to Aux2 24:Move disk 3 from Aux3 to Aux4 25:Move disk 1 from Aux2 to Aux3 26:Move disk 1 from Aux3 to Aux4 27:Move disk 2 from Aux2 to Aux3 28:Move disk 1 from Aux4 to Aux1 29:Move disk 1 from Aux1 to Aux2 30:Move disk 2 from Aux3 to Aux4 31:Move disk 1 from Aux2 to Aux3 32:Move disk 1 from Aux3 to Aux4 33:Move disk 4 from Start to Aux1 34:Move disk 4 from Aux1 to Aux2 35:Move disk 4 from Aux2 to Aux3 36:Move disk 1 from Aux4 to Aux1 37:Move disk 1 from Aux1 to Aux2

38:Move disk 2 from Aux4 to Aux1	38:Move disk 2 from Aux4 to Aux1	38:Move disk 2 from Aux4 to Aux1	38:Move disk 2 from Aux4 to Aux1
39:Move disk 1 from Aux2 to Aux3	39:Move disk 1 from Aux2 to Aux3	39:Move disk 1 from Aux2 to Aux3	39:Move disk 1 from Aux2 to Aux3
40:Move disk 1 from Aux3 to Aux4	40:Move disk 1 from Aux3 to Aux4	40:Move disk 1 from Aux3 to Aux4	40:Move disk 1 from Aux3 to Aux4
41:Move disk 2 from Aux1 to Aux2	41:Move disk 2 from Aux1 to Aux2	41:Move disk 2 from Aux1 to Aux2	41:Move disk 2 from Aux1 to Aux2
42:Move disk 1 from Aux4 to Aux1	42:Move disk 1 from Aux4 to Aux1	42:Move disk 1 from Aux4 to Aux1	42:Move disk 1 from Aux4 to Aux1
43:Move disk 1 from Aux1 to Aux2	43:Move disk 1 from Aux1 to Aux2	43:Move disk 1 from Aux1 to Aux2	43:Move disk 1 from Aux1 to Aux2
44:Move disk 3 from Aux4 to Aux1	44:Move disk 3 from Aux4 to Aux1	44:Move disk 3 from Aux4 to Aux1	44:Move disk 3 from Aux4 to Aux1
45:Move disk 1 from Aux2 to Aux3	45:Move disk 1 from Aux2 to Aux3	45:Move disk 1 from Aux2 to Aux3	45:Move disk 1 from Aux2 to Aux3
46:Move disk 1 from Aux3 to Aux4	46:Move disk 1 from Aux3 to Aux4	46:Move disk 1 from Aux3 to Aux4	46:Move disk 1 from Aux3 to Aux4
47:Move disk 2 from Aux2 to Aux3	47:Move disk 2 from Aux2 to Aux3	47:Move disk 2 from Aux2 to Aux3	47:Move disk 2 from Aux2 to Aux3
48:Move disk 1 from Aux4 to Aux1	48:Move disk 1 from Aux4 to Aux1	48:Move disk 1 from Aux4 to Aux1	48:Move disk 1 from Aux4 to Aux1
49:Move disk 1 from Aux1 to Aux2	49:Move disk 1 from Aux1 to Aux2	49:Move disk 1 from Aux1 to Aux2	49:Move disk 1 from Aux1 to Aux2
50:Move disk 2 from Aux3 to Aux4	50:Move disk 2 from Aux3 to Aux4	50:Move disk 2 from Aux3 to Aux4	50:Move disk 2 from Aux3 to Aux4
51:Move disk 1 from Aux2 to Aux3	51:Move disk 1 from Aux2 to Aux3	51:Move disk 1 from Aux2 to Aux3	51:Move disk 1 from Aux2 to Aux3
52:Move disk 1 from Aux3 to Aux4	52:Move disk 1 from Aux3 to Aux4	52:Move disk 1 from Aux3 to Aux4	52:Move disk 1 from Aux3 to Aux4
53:Move disk 3 from Aux1 to Aux2	53:Move disk 3 from Aux1 to Aux2	53:Move disk 3 from Aux1 to Aux2	53:Move disk 3 from Aux1 to Aux2
54:Move disk 1 from Aux4 to Aux1	54:Move disk 1 from Aux4 to Aux1	54:Move disk 1 from Aux4 to Aux1	54:Move disk 1 from Aux4 to Aux1
55:Move disk 1 from Aux1 to Aux2	55:Move disk 1 from Aux1 to Aux2	55:Move disk 1 from Aux1 to Aux2	55:Move disk 1 from Aux1 to Aux2
56:Move disk 2 from Aux4 to Aux1	56:Move disk 2 from Aux4 to Aux1	56:Move disk 2 from Aux4 to Aux1	56:Move disk 2 from Aux4 to Aux1
57:Move disk 1 from Aux2 to Aux3	57:Move disk 1 from Aux2 to Aux3	57:Move disk 1 from Aux2 to Aux3	57:Move disk 1 from Aux2 to Aux3
58:Move disk 1 from Aux3 to Aux4	58:Move disk 1 from Aux3 to Aux4	58:Move disk 1 from Aux3 to Aux4	58:Move disk 1 from Aux3 to Aux4
59:Move disk 2 from Aux1 to Aux2	59:Move disk 2 from Aux1 to Aux2	59:Move disk 2 from Aux1 to Aux2	59:Move disk 2 from Aux1 to Aux2
60:Move disk 1 from Aux4 to Aux1	60:Move disk 1 from Aux4 to Aux1	60:Move disk 1 from Aux4 to Aux1	60:Move disk 1 from Aux4 to Aux1
61:Move disk 1 from Aux1 to Aux2	61:Move disk 1 from Aux1 to Aux2	61:Move disk 1 from Aux1 to Aux2	61:Move disk 1 from Aux1 to Aux2
62:Move disk 4 from Aux3 to Aux4	62:Move disk 4 from Aux3 to Aux4	62:Move disk 4 from Aux3 to Aux4	62:Move disk 4 from Aux3 to Aux4
63:Move disk 1 from Aux2 to Aux3	63:Move disk 1 from Aux2 to Aux3	63:Move disk 1 from Aux2 to Aux3	63:Move disk 1 from Aux2 to Aux3
64:Move disk 1 from Aux3 to Aux4	64:Move disk 1 from Aux3 to Aux4	64:Move disk 1 from Aux3 to Aux4	64:Move disk 1 from Aux3 to Aux4
65:Move disk 2 from Aux2 to Aux3	65:Move disk 2 from Aux2 to Aux3	65:Move disk 2 from Aux2 to Aux3	65:Move disk 2 from Aux2 to Aux3
66:Move disk 1 from Aux4 to Aux1	66:Move disk 1 from Aux4 to Aux1	66:Move disk 1 from Aux4 to Aux1	66:Move disk 1 from Aux4 to Aux1
67:Move disk 1 from Aux1 to Aux2	67:Move disk 1 from Aux1 to Aux2	67:Move disk 1 from Aux1 to Aux2	67:Move disk 1 from Aux1 to Aux2
68:Move disk 2 from Aux3	68:Move disk 2 from Aux3	68:Move disk 2 from Aux3	68:Move disk 2 from Aux3

to Aux4	to Aux4	to Aux4	to Aux4
69:Move disk 1 from Aux2 to Aux3	69:Move disk 1 from Aux2 to Aux3	69:Move disk 1 from Aux2 to Aux3	69:Move disk 1 from Aux2 to Aux3
70:Move disk 1 from Aux3 to Aux4	70:Move disk 1 from Aux3 to Aux4	70:Move disk 1 from Aux3 to Aux4	70:Move disk 1 from Aux3 to Aux4
71:Move disk 3 from Aux2 to Aux3	71:Move disk 3 from Aux2 to Aux3	71:Move disk 3 from Aux2 to Aux3	71:Move disk 3 from Aux2 to Aux3
72:Move disk 1 from Aux4 to Aux1	72:Move disk 1 from Aux4 to Aux1	72:Move disk 1 from Aux4 to Aux1	72:Move disk 1 from Aux4 to Aux1
73:Move disk 1 from Aux1 to Aux2	73:Move disk 1 from Aux1 to Aux2	73:Move disk 1 from Aux1 to Aux2	73:Move disk 1 from Aux1 to Aux2
74:Move disk 2 from Aux4 to Aux1	74:Move disk 2 from Aux4 to Aux1	74:Move disk 2 from Aux4 to Aux1	74:Move disk 2 from Aux4 to Aux1
75:Move disk 1 from Aux2 to Aux3	75:Move disk 1 from Aux2 to Aux3	75:Move disk 1 from Aux2 to Aux3	75:Move disk 1 from Aux2 to Aux3
76:Move disk 1 from Aux3 to Aux4	76:Move disk 1 from Aux3 to Aux4	76:Move disk 1 from Aux3 to Aux4	76:Move disk 1 from Aux3 to Aux4
77:Move disk 2 from Aux1 to Aux2	77:Move disk 2 from Aux1 to Aux2	77:Move disk 2 from Aux1 to Aux2	77:Move disk 2 from Aux1 to Aux2
78:Move disk 1 from Aux4 to Aux1	78:Move disk 1 from Aux4 to Aux1	78:Move disk 1 from Aux4 to Aux1	78:Move disk 1 from Aux4 to Aux1
79:Move disk 1 from Aux1 to Aux2	79:Move disk 1 from Aux1 to Aux2	79:Move disk 1 from Aux1 to Aux2	79:Move disk 1 from Aux1 to Aux2
80:Move disk 3 from Aux3 to Aux4	80:Move disk 3 from Aux3 to Aux4	80:Move disk 3 from Aux3 to Aux4	80:Move disk 3 from Aux3 to Aux4
81:Move disk 1 from Aux2 to Aux3	81:Move disk 1 from Aux2 to Aux3	81:Move disk 1 from Aux2 to Aux3	81:Move disk 1 from Aux2 to Aux3
82:Move disk 1 from Aux3 to Aux4	82:Move disk 1 from Aux3 to Aux4	82:Move disk 1 from Aux3 to Aux4	82:Move disk 1 from Aux3 to Aux4
83:Move disk 2 from Aux2 to Aux3	83:Move disk 2 from Aux2 to Aux3	83:Move disk 2 from Aux2 to Aux3	83:Move disk 2 from Aux2 to Aux3
84:Move disk 1 from Aux4 to Aux1	84:Move disk 1 from Aux4 to Aux1	84:Move disk 1 from Aux4 to Aux1	84:Move disk 1 from Aux4 to Aux1
85:Move disk 1 from Aux1 to Aux2	85:Move disk 1 from Aux1 to Aux2	85:Move disk 1 from Aux1 to Aux2	85:Move disk 1 from Aux1 to Aux2
86:Move disk 2 from Aux3 to Aux4	86:Move disk 2 from Aux3 to Aux4	86:Move disk 2 from Aux3 to Aux4	86:Move disk 2 from Aux3 to Aux4
87:Move disk 1 from Aux2 to Aux3	87:Move disk 1 from Aux2 to Aux3	87:Move disk 1 from Aux2 to Aux3	87:Move disk 1 from Aux2 to Aux3
88:Move disk 1 from Aux3 to Aux4	88:Move disk 1 from Aux3 to Aux4	88:Move disk 1 from Aux3 to Aux4	88:Move disk 1 from Aux3 to Aux4
89:Move disk 5 from Start to Aux1	89:Move disk 5 from Start to Aux1	89:Move disk 5 from Start to Aux1	89:Move disk 5 from Start to Aux1
90:Move disk 5 from Aux1 to Aux2	90:Move disk 5 from Aux1 to Aux2	90:Move disk 5 from Aux1 to Aux2	90:Move disk 5 from Aux1 to Aux2
91:Move disk 5 from Aux2 to Aux3	91:Move disk 5 from Aux2 to Aux3	91:Move disk 5 from Aux2 to Aux3	91:Move disk 5 from Aux2 to Aux3
92:Move disk 5 from Aux3 to Dest	92:Move disk 1 from Aux4 to Aux1	92:Move disk 1 from Aux4 to Aux1	92:Move disk 1 from Aux4 to Aux1
93:Move disk 1 from Aux4 to Aux1	93:Move disk 1 from Aux1 to Aux2	93:Move disk 1 from Aux1 to Aux2	93:Move disk 1 from Aux1 to Aux2
94:Move disk 1 from Aux1 to Aux2	94:Move disk 2 from Aux4 to Aux1	94:Move disk 2 from Aux4 to Aux1	94:Move disk 2 from Aux4 to Aux1
95:Move disk 2 from Aux4 to Aux1	95:Move disk 1 from Aux2 to Aux3	95:Move disk 1 from Aux2 to Aux3	95:Move disk 1 from Aux2 to Aux3
96:Move disk 1 from Aux2 to Aux3	96:Move disk 1 from Aux3 to Aux4	96:Move disk 1 from Aux3 to Aux4	96:Move disk 1 from Aux3 to Aux4
97:Move disk 1 from Aux3 to Aux4	97:Move disk 2 from Aux1 to Aux2	97:Move disk 2 from Aux1 to Aux2	97:Move disk 2 from Aux1 to Aux2
98:Move disk 2 from Aux1 to Aux2	98:Move disk 1 from Aux4 to Aux1	98:Move disk 1 from Aux4 to Aux1	98:Move disk 1 from Aux4 to Aux1

99:Move disk 1 from Aux4 to Aux1 100:Move disk 1 from Aux1 to Aux2 101:Move disk 3 from Aux4 to Aux1 102:Move disk 1 from Aux2 to Aux3 103:Move disk 1 from Aux3 to Aux4 104:Move disk 2 from Aux2 to Aux3 105:Move disk 1 from Aux4 to Aux1 106:Move disk 1 from Aux1 to Aux2 107:Move disk 2 from Aux3 to Aux4 108:Move disk 1 from Aux2 to Aux3 109:Move disk 1 from Aux3 to Aux4 110:Move disk 3 from Aux1 to Aux2 111:Move disk 1 from Aux4 to Aux1 112:Move disk 1 from Aux1 to Aux2 113:Move disk 2 from Aux4 to Aux1 114:Move disk 1 from Aux2 to Aux3 115:Move disk 1 from Aux3 to Aux4 116:Move disk 2 from Aux1 to Aux2 117:Move disk 1 from Aux4 to Aux1 118:Move disk 1 from Aux1 to Aux2 119:Move disk 4 from Aux4 to Aux1 120:Move disk 1 from Aux2 to Aux3 121:Move disk 1 from Aux3 to Aux4 122:Move disk 2 from Aux2 to Aux3 123:Move disk 1 from Aux4 to Aux1 124:Move disk 1 from Aux1 to Aux2 125:Move disk 2 from Aux3 to Aux4 126:Move disk 1 from Aux2 to Aux3 127:Move disk 1 from Aux3 to Aux4 128:Move disk 3 from Aux2 to Aux3 129:Move disk 1 from	99:Move disk 1 from Aux1 to Aux2 100:Move disk 3 from Aux4 to Aux1 **LAST 100** 409:Move disk 3 from Aux3 to Aux4 410:Move disk 1 from Aux2 to Aux3 411:Move disk 1 from Aux3 to Aux4 412:Move disk 2 from Aux2 to Aux3 413:Move disk 1 from Aux4 to Aux1 414:Move disk 1 from Aux1 to Aux2 415:Move disk 2 from Aux3 to Aux4 416:Move disk 1 from Aux2 to Aux3 417:Move disk 1 from Aux3 to Aux4 418:Move disk 5 from Aux1 to Aux2 419:Move disk 5 from Aux2 to Aux3 420:Move disk 5 from Aux3 to Dest 421:Move disk 1 from Aux4 to Aux1 422:Move disk 1 from Aux1 to Aux2 423:Move disk 2 from Aux4 to Aux1 424:Move disk 1 from Aux2 to Aux3 425:Move disk 1 from Aux3 to Aux4 426:Move disk 2 from Aux1 to Aux2 427:Move disk 1 from Aux4 to Aux1 428:Move disk 1 from Aux1 to Aux2 429:Move disk 3 from Aux4 to Aux1 430:Move disk 1 from Aux2 to Aux3 431:Move disk 1 from Aux3 to Aux4 432:Move disk 2 from Aux2 to Aux3 433:Move disk 1 from Aux4 to Aux1 434:Move disk 1 from Aux1 to Aux2 435:Move disk 2 from Aux3 to Aux4 436:Move disk 1 from Aux2 to Aux3	99:Move disk 1 from Aux1 to Aux2 100:Move disk 3 from Aux4 to Aux1 **LAST 100** 1385:Move disk 3 from Aux3 to Aux4 1386:Move disk 1 from Aux2 to Aux3 1387:Move disk 1 from Aux3 to Aux4 1388:Move disk 2 from Aux2 to Aux3 1389:Move disk 1 from Aux4 to Aux1 1390:Move disk 1 from Aux1 to Aux2 1391:Move disk 2 from Aux3 to Aux4 1392:Move disk 1 from Aux2 to Aux3 1393:Move disk 1 from Aux3 to Aux4 1394:Move disk 5 from Aux1 to Aux2 1395:Move disk 5 from Aux2 to Aux3 1396:Move disk 5 from Aux3 to Dest 1397:Move disk 1 from Aux4 to Aux1 1398:Move disk 1 from Aux1 to Aux2 1399:Move disk 2 from Aux4 to Aux1 1400:Move disk 1 from Aux2 to Aux3 1401:Move disk 1 from Aux3 to Aux4 1402:Move disk 2 from Aux1 to Aux2 1403:Move disk 1 from Aux4 to Aux1 1404:Move disk 1 from Aux1 to Aux2 1405:Move disk 3 from Aux4 to Aux1 1406:Move disk 1 from Aux2 to Aux3 1407:Move disk 1 from Aux3 to Aux4 1408:Move disk 2 from Aux2 to Aux3 1409:Move disk 1 from Aux4 to Aux1 1410:Move disk 1 from Aux1 to Aux2 1411:Move disk 2 from Aux3 to Aux4 1412:Move disk 1 from Aux2 to Aux3	99:Move disk 1 from Aux1 to Aux2 100:Move disk 3 from Aux4 to Aux1 **LAST 100** 4305:Move disk 3 from Aux3 to Aux4 4306:Move disk 1 from Aux2 to Aux3 4307:Move disk 1 from Aux3 to Aux4 4308:Move disk 2 from Aux2 to Aux3 4309:Move disk 1 from Aux4 to Aux1 4310:Move disk 1 from Aux1 to Aux2 4311:Move disk 2 from Aux3 to Aux4 4312:Move disk 1 from Aux2 to Aux3 4313:Move disk 1 from Aux3 to Aux4 4314:Move disk 5 from Aux1 to Aux2 4315:Move disk 5 from Aux2 to Aux3 4316:Move disk 5 from Aux3 to Dest 4317:Move disk 1 from Aux4 to Aux1 4318:Move disk 1 from Aux1 to Aux2 4319:Move disk 2 from Aux4 to Aux1 4320:Move disk 1 from Aux2 to Aux3 4321:Move disk 1 from Aux3 to Aux4 4322:Move disk 2 from Aux1 to Aux2 4323:Move disk 1 from Aux4 to Aux1 4324:Move disk 1 from Aux1 to Aux2 4325:Move disk 3 from Aux4 to Aux1 4326:Move disk 1 from Aux2 to Aux3 4327:Move disk 1 from Aux3 to Aux4 4328:Move disk 2 from Aux2 to Aux3 4329:Move disk 1 from Aux4 to Aux1 4330:Move disk 1 from Aux1 to Aux2 4331:Move disk 2 from Aux3 to Aux4 4332:Move disk 1 from Aux2 to Aux3

Aux4 to Aux1
130:Move disk 1 from Aux1 to Aux2
131:Move disk 2 from Aux4 to Aux1
132:Move disk 1 from Aux2 to Aux3
133:Move disk 1 from Aux3 to Aux4
134:Move disk 2 from Aux1 to Aux2
135:Move disk 1 from Aux4 to Aux1
136:Move disk 1 from Aux1 to Aux2
137:Move disk 3 from Aux3 to Aux4
138:Move disk 1 from Aux2 to Aux3
139:Move disk 1 from Aux3 to Aux4
140:Move disk 2 from Aux2 to Aux3
141:Move disk 1 from Aux4 to Aux1
142:Move disk 1 from Aux1 to Aux2
143:Move disk 2 from Aux3 to Aux4
144:Move disk 1 from Aux2 to Aux3
145:Move disk 1 from Aux3 to Aux4
146:Move disk 4 from Aux1 to Aux2
147:Move disk 4 from Aux2 to Aux3
148:Move disk 4 from Aux3 to Dest
149:Move disk 1 from Aux4 to Aux1
150:Move disk 1 from Aux1 to Aux2
151:Move disk 2 from Aux4 to Aux1
152:Move disk 1 from Aux2 to Aux3
153:Move disk 1 from Aux3 to Aux4
154:Move disk 2 from Aux1 to Aux2
155:Move disk 1 from Aux4 to Aux1
156:Move disk 1 from Aux1 to Aux2
157:Move disk 3 from Aux4 to Aux1
158:Move disk 1 from Aux2 to Aux3
159:Move disk 1 from Aux3 to Aux4
437:Move disk 1 from Aux3 to Aux4
438:Move disk 3 from Aux1 to Aux2
439:Move disk 1 from Aux4 to Aux1
440:Move disk 1 from Aux1 to Aux2
441:Move disk 2 from Aux4 to Aux1
442:Move disk 1 from Aux2 to Aux3
443:Move disk 1 from Aux3 to Aux4
444:Move disk 2 from Aux1 to Aux2
445:Move disk 1 from Aux4 to Aux1
446:Move disk 1 from Aux1 to Aux2
447:Move disk 4 from Aux4 to Aux1
448:Move disk 1 from Aux2 to Aux3
449:Move disk 1 from Aux3 to Aux4
450:Move disk 2 from Aux2 to Aux3
451:Move disk 1 from Aux4 to Aux1
452:Move disk 1 from Aux1 to Aux2
453:Move disk 2 from Aux3 to Aux4
454:Move disk 1 from Aux2 to Aux3
455:Move disk 1 from Aux3 to Aux4
456:Move disk 3 from Aux2 to Aux3
457:Move disk 1 from Aux4 to Aux1
458:Move disk 1 from Aux1 to Aux2
459:Move disk 2 from Aux4 to Aux1
460:Move disk 1 from Aux2 to Aux3
461:Move disk 1 from Aux3 to Aux4
462:Move disk 2 from Aux1 to Aux2
463:Move disk 1 from Aux4 to Aux1
464:Move disk 1 from Aux1 to Aux2
465:Move disk 3 from Aux3 to Aux4
466:Move disk 1 from Aux2 to Aux3
467:Move disk 1 from
1413:Move disk 1 from Aux3 to Aux4
1414:Move disk 3 from Aux1 to Aux2
1415:Move disk 1 from Aux4 to Aux1
1416:Move disk 1 from Aux1 to Aux2
1417:Move disk 2 from Aux4 to Aux1
1418:Move disk 1 from Aux2 to Aux3
1419:Move disk 1 from Aux3 to Aux4
1420:Move disk 2 from Aux1 to Aux2
1421:Move disk 1 from Aux4 to Aux1
1422:Move disk 1 from Aux1 to Aux2
1423:Move disk 4 from Aux4 to Aux1
1424:Move disk 1 from Aux2 to Aux3
1425:Move disk 1 from Aux3 to Aux4
1426:Move disk 2 from Aux2 to Aux3
1427:Move disk 1 from Aux4 to Aux1
1428:Move disk 1 from Aux1 to Aux2
1429:Move disk 2 from Aux3 to Aux4
1430:Move disk 1 from Aux2 to Aux3
1431:Move disk 1 from Aux3 to Aux4
1432:Move disk 3 from Aux2 to Aux3
1433:Move disk 1 from Aux4 to Aux1
1434:Move disk 1 from Aux1 to Aux2
1435:Move disk 2 from Aux4 to Aux1
1436:Move disk 1 from Aux2 to Aux3
1437:Move disk 1 from Aux3 to Aux4
1438:Move disk 2 from Aux1 to Aux2
1439:Move disk 1 from Aux4 to Aux1
1440:Move disk 1 from Aux1 to Aux2
1441:Move disk 3 from Aux3 to Aux4
1442:Move disk 1 from Aux2 to Aux3
1443:Move disk 1 from
4333:Move disk 1 from Aux3 to Aux4
4334:Move disk 3 from Aux1 to Aux2
4335:Move disk 1 from Aux4 to Aux1
4336:Move disk 1 from Aux1 to Aux2
4337:Move disk 2 from Aux4 to Aux1
4338:Move disk 1 from Aux2 to Aux3
4339:Move disk 1 from Aux3 to Aux4
4340:Move disk 2 from Aux1 to Aux2
4341:Move disk 1 from Aux4 to Aux1
4342:Move disk 1 from Aux1 to Aux2
4343:Move disk 4 from Aux4 to Aux1
4344:Move disk 1 from Aux2 to Aux3
4345:Move disk 1 from Aux3 to Aux4
4346:Move disk 2 from Aux2 to Aux3
4347:Move disk 1 from Aux4 to Aux1
4348:Move disk 1 from Aux1 to Aux2
4349:Move disk 2 from Aux3 to Aux4
4350:Move disk 1 from Aux2 to Aux3
4351:Move disk 1 from Aux3 to Aux4
4352:Move disk 3 from Aux2 to Aux3
4353:Move disk 1 from Aux4 to Aux1
4354:Move disk 1 from Aux1 to Aux2
4355:Move disk 2 from Aux4 to Aux1
4356:Move disk 1 from Aux2 to Aux3
4357:Move disk 1 from Aux3 to Aux4
4358:Move disk 2 from Aux1 to Aux2
4359:Move disk 1 from Aux4 to Aux1
4360:Move disk 1 from Aux1 to Aux2
4361:Move disk 3 from Aux3 to Aux4
4362:Move disk 1 from Aux2 to Aux3
4363:Move disk 1 from

160:Move disk 2 from Aux2 to Aux3
161:Move disk 1 from Aux4 to Aux1
162:Move disk 1 from Aux1 to Aux2
163:Move disk 2 from Aux3 to Aux4
164:Move disk 1 from Aux2 to Aux3
165:Move disk 1 from Aux3 to Aux4
166:Move disk 3 from Aux1 to Aux2
167:Move disk 3 from Aux2 to Aux3
168:Move disk 3 from Aux3 to Dest
169:Move disk 1 from Aux4 to Aux1
170:Move disk 1 from Aux1 to Aux2
171:Move disk 2 from Aux4 to Aux1
172:Move disk 1 from Aux2 to Aux3
173:Move disk 1 from Aux3 to Aux4
174:Move disk 2 from Aux1 to Aux2
175:Move disk 2 from Aux2 to Aux3
176:Move disk 2 from Aux3 to Dest
177:Move disk 1 from Aux4 to Aux1
178:Move disk 1 from Aux1 to Aux2
179:Move disk 1 from Aux2 to Aux3
180:Move disk 1 from Aux3 to Dest

Aux3 to Aux4
468:Move disk 2 from Aux2 to Aux3
469:Move disk 1 from Aux4 to Aux1
470:Move disk 1 from Aux1 to Aux2
471:Move disk 2 from Aux3 to Aux4
472:Move disk 1 from Aux2 to Aux3
473:Move disk 1 from Aux3 to Aux4
474:Move disk 4 from Aux1 to Aux2
475:Move disk 4 from Aux2 to Aux3
476:Move disk 4 from Aux3 to Dest
477:Move disk 1 from Aux4 to Aux1
478:Move disk 1 from Aux1 to Aux2
479:Move disk 2 from Aux4 to Aux1
480:Move disk 1 from Aux2 to Aux3
481:Move disk 1 from Aux3 to Aux4
482:Move disk 2 from Aux1 to Aux2
483:Move disk 1 from Aux4 to Aux1
484:Move disk 1 from Aux1 to Aux2
485:Move disk 3 from Aux4 to Aux1
486:Move disk 1 from Aux2 to Aux3
487:Move disk 1 from Aux3 to Aux4
488:Move disk 2 from Aux2 to Aux3
489:Move disk 1 from Aux4 to Aux1
490:Move disk 1 from Aux1 to Aux2
491:Move disk 2 from Aux3 to Aux4
492:Move disk 1 from Aux2 to Aux3
493:Move disk 1 from Aux3 to Aux4
494:Move disk 3 from Aux1 to Aux2
495:Move disk 3 from Aux2 to Aux3
496:Move disk 3 from Aux3 to Dest
497:Move disk 1 from Aux4 to Aux1

Aux3 to Aux4
1444:Move disk 2 from Aux2 to Aux3
1445:Move disk 1 from Aux4 to Aux1
1446:Move disk 1 from Aux1 to Aux2
1447:Move disk 2 from Aux3 to Aux4
1448:Move disk 1 from Aux2 to Aux3
1449:Move disk 1 from Aux3 to Aux4
1450:Move disk 4 from Aux1 to Aux2
1451:Move disk 4 from Aux2 to Aux3
1452:Move disk 4 from Aux3 to Dest
1453:Move disk 1 from Aux4 to Aux1
1454:Move disk 1 from Aux1 to Aux2
1455:Move disk 2 from Aux4 to Aux1
1456:Move disk 1 from Aux2 to Aux3
1457:Move disk 1 from Aux3 to Aux4
1458:Move disk 2 from Aux1 to Aux2
1459:Move disk 1 from Aux4 to Aux1
1460:Move disk 1 from Aux1 to Aux2
1461:Move disk 3 from Aux4 to Aux1
1462:Move disk 1 from Aux2 to Aux3
1463:Move disk 1 from Aux3 to Aux4
1464:Move disk 2 from Aux2 to Aux3
1465:Move disk 1 from Aux4 to Aux1
1466:Move disk 1 from Aux1 to Aux2
1467:Move disk 2 from Aux3 to Aux4
1468:Move disk 1 from Aux2 to Aux3
1469:Move disk 1 from Aux3 to Aux4
1470:Move disk 3 from Aux1 to Aux2
1471:Move disk 3 from Aux2 to Aux3
1472:Move disk 3 from Aux3 to Dest
1473:Move disk 1 from Aux4 to Aux1

Aux3 to Aux4
4364:Move disk 2 from Aux2 to Aux3
4365:Move disk 1 from Aux4 to Aux1
4366:Move disk 1 from Aux1 to Aux2
4367:Move disk 2 from Aux3 to Aux4
4368:Move disk 1 from Aux2 to Aux3
4369:Move disk 1 from Aux3 to Aux4
4370:Move disk 4 from Aux1 to Aux2
4371:Move disk 4 from Aux2 to Aux3
4372:Move disk 4 from Aux3 to Dest
4373:Move disk 1 from Aux4 to Aux1
4374:Move disk 1 from Aux1 to Aux2
4375:Move disk 2 from Aux4 to Aux1
4376:Move disk 1 from Aux2 to Aux3
4377:Move disk 1 from Aux3 to Aux4
4378:Move disk 2 from Aux1 to Aux2
4379:Move disk 1 from Aux4 to Aux1
4380:Move disk 1 from Aux1 to Aux2
4381:Move disk 3 from Aux4 to Aux1
4382:Move disk 1 from Aux2 to Aux3
4383:Move disk 1 from Aux3 to Aux4
4384:Move disk 2 from Aux2 to Aux3
4385:Move disk 1 from Aux4 to Aux1
4386:Move disk 1 from Aux1 to Aux2
4387:Move disk 2 from Aux3 to Aux4
4388:Move disk 1 from Aux2 to Aux3
4389:Move disk 1 from Aux3 to Aux4
4390:Move disk 3 from Aux1 to Aux2
4391:Move disk 3 from Aux2 to Aux3
4392:Move disk 3 from Aux3 to Dest
4393:Move disk 1 from Aux4 to Aux1

	498:Move disk 1 from Aux1 to Aux2 499:Move disk 2 from Aux4 to Aux1 500:Move disk 1 from Aux2 to Aux3 501:Move disk 1 from Aux3 to Aux4 502:Move disk 2 from Aux1 to Aux2 503:Move disk 2 from Aux2 to Aux3 504:Move disk 2 from Aux3 to Dest 505:Move disk 1 from Aux4 to Aux1 506:Move disk 1 from Aux1 to Aux2 507:Move disk 1 from Aux2 to Aux3 508:Move disk 1 from Aux3 to Dest	1474:Move disk 1 from Aux1 to Aux2 1475:Move disk 2 from Aux4 to Aux1 1476:Move disk 1 from Aux2 to Aux3 1477:Move disk 1 from Aux3 to Aux4 1478:Move disk 2 from Aux1 to Aux2 1479:Move disk 2 from Aux2 to Aux3 1480:Move disk 2 from Aux3 to Dest 1481:Move disk 1 from Aux4 to Aux1 1482:Move disk 1 from Aux1 to Aux2 1483:Move disk 1 from Aux2 to Aux3 1484:Move disk 1 from Aux3 to Dest	4394:Move disk 1 from Aux1 to Aux2 4395:Move disk 2 from Aux4 to Aux1 4396:Move disk 1 from Aux2 to Aux3 4397:Move disk 1 from Aux3 to Aux4 4398:Move disk 2 from Aux1 to Aux2 4399:Move disk 2 from Aux2 to Aux3 4400:Move disk 2 from Aux3 to Dest 4401:Move disk 1 from Aux4 to Aux1 4402:Move disk 1 from Aux1 to Aux2 4403:Move disk 1 from Aux2 to Aux3 4404:Move disk 1 from Aux3 to Dest

n=9 1:Move disk 1 from Start to Aux1 2:Move disk 1 from Aux1 to Aux2 3:Move disk 1 from Aux2 to Aux3 4:Move disk 1 from Aux3 to Aux4 5:Move disk 2 from Start to Aux1 6:Move disk 2 from Aux1 to Aux2 7:Move disk 2 from Aux2 to Aux3 8:Move disk 1 from Aux4 to Aux1 9:Move disk 1 from Aux1 to Aux2 10:Move disk 2 from Aux3 to Aux4 11:Move disk 1 from Aux2 to Aux3 12:Move disk 1 from Aux3 to Aux4 13:Move disk 3 from Start to Aux1 14:Move disk 3 from Aux1 to Aux2 15:Move disk 3 from Aux2 to Aux3 16:Move disk 1 from Aux4 to Aux1 17:Move disk 1 from Aux1 to Aux2 18:Move disk 2 from Aux4 to Aux1 19:Move disk 1 from Aux2 to Aux3 20:Move disk 1 from Aux3 to Aux4 21:Move disk 2 from Aux1 to Aux2 22:Move disk 1 from Aux4 to Aux1 23:Move disk 1 from Aux1 to Aux2 24:Move disk 3 from Aux3 to Aux4 25:Move disk 1 from Aux2 to Aux3 26:Move disk 1 from Aux3 to Aux4 27:Move disk 2 from Aux2 to Aux3 28:Move disk 1 from Aux4 to Aux1 29:Move disk 1 from Aux1 to Aux2 30:Move disk 2 from Aux3 to Aux4	n=10 1:Move disk 1 from Start to Aux1 2:Move disk 1 from Aux1 to Aux2 3:Move disk 1 from Aux2 to Aux3 4:Move disk 1 from Aux3 to Aux4 5:Move disk 2 from Start to Aux1 6:Move disk 2 from Aux1 to Aux2 7:Move disk 2 from Aux2 to Aux3 8:Move disk 1 from Aux4 to Aux1 9:Move disk 1 from Aux1 to Aux2 10:Move disk 2 from Aux3 to Aux4 11:Move disk 1 from Aux2 to Aux3 12:Move disk 1 from Aux3 to Aux4 13:Move disk 3 from Start to Aux1 14:Move disk 3 from Aux1 to Aux2 15:Move disk 3 from Aux2 to Aux3 16:Move disk 1 from Aux4 to Aux1 17:Move disk 1 from Aux1 to Aux2 18:Move disk 2 from Aux4 to Aux1 19:Move disk 1 from Aux2 to Aux3 20:Move disk 1 from Aux3 to Aux4 21:Move disk 2 from Aux1 to Aux2 22:Move disk 1 from Aux4 to Aux1 23:Move disk 1 from Aux1 to Aux2 24:Move disk 3 from Aux3 to Aux4 25:Move disk 1 from Aux2 to Aux3 26:Move disk 1 from Aux3 to Aux4 27:Move disk 2 from Aux2 to Aux3 28:Move disk 1 from Aux4 to Aux1 29:Move disk 1 from Aux1 to Aux2 30:Move disk 2 from Aux3 to Aux4		

31:Move disk 1 from Aux2 to Aux3 32:Move disk 1 from Aux3 to Aux4 33:Move disk 4 from Start to Aux1 34:Move disk 4 from Aux1 to Aux2 35:Move disk 4 from Aux2 to Aux3 36:Move disk 1 from Aux4 to Aux1 37:Move disk 1 from Aux1 to Aux2 38:Move disk 2 from Aux4 to Aux1 39:Move disk 1 from Aux2 to Aux3 40:Move disk 1 from Aux3 to Aux4 41:Move disk 2 from Aux1 to Aux2 42:Move disk 1 from Aux4 to Aux1 43:Move disk 1 from Aux1 to Aux2 44:Move disk 3 from Aux4 to Aux1 45:Move disk 1 from Aux2 to Aux3 46:Move disk 1 from Aux3 to Aux4 47:Move disk 2 from Aux2 to Aux3 48:Move disk 1 from Aux4 to Aux1 49:Move disk 1 from Aux1 to Aux2 50:Move disk 2 from Aux3 to Aux4 51:Move disk 1 from Aux2 to Aux3 52:Move disk 1 from Aux3 to Aux4 53:Move disk 3 from Aux1 to Aux2 54:Move disk 1 from Aux4 to Aux1 55:Move disk 1 from Aux1 to Aux2 56:Move disk 2 from Aux4 to Aux1 57:Move disk 1 from Aux2 to Aux3 58:Move disk 1 from Aux3 to Aux4 59:Move disk 2 from Aux1 to Aux2 60:Move disk 1 from Aux4 to Aux1 61:Move disk 1 from Aux1	31:Move disk 1 from Aux2 to Aux3 32:Move disk 1 from Aux3 to Aux4 33:Move disk 4 from Start to Aux1 34:Move disk 4 from Aux1 to Aux2 35:Move disk 4 from Aux2 to Aux3 36:Move disk 1 from Aux4 to Aux1 37:Move disk 1 from Aux1 to Aux2 38:Move disk 2 from Aux4 to Aux1 39:Move disk 1 from Aux2 to Aux3 40:Move disk 1 from Aux3 to Aux4 41:Move disk 2 from Aux1 to Aux2 42:Move disk 1 from Aux4 to Aux1 43:Move disk 1 from Aux1 to Aux2 44:Move disk 3 from Aux4 to Aux1 45:Move disk 1 from Aux2 to Aux3 46:Move disk 1 from Aux3 to Aux4 47:Move disk 2 from Aux2 to Aux3 48:Move disk 1 from Aux4 to Aux1 49:Move disk 1 from Aux1 to Aux2 50:Move disk 2 from Aux3 to Aux4 51:Move disk 1 from Aux2 to Aux3 52:Move disk 1 from Aux3 to Aux4 53:Move disk 3 from Aux1 to Aux2 54:Move disk 1 from Aux4 to Aux1 55:Move disk 1 from Aux1 to Aux2 56:Move disk 2 from Aux4 to Aux1 57:Move disk 1 from Aux2 to Aux3 58:Move disk 1 from Aux3 to Aux4 59:Move disk 2 from Aux1 to Aux2 60:Move disk 1 from Aux4 to Aux1 61:Move disk 1 from Aux1		

to Aux2 62:Move disk 4 from Aux3 to Aux4 63:Move disk 1 from Aux2 to Aux3 64:Move disk 1 from Aux3 to Aux4 65:Move disk 2 from Aux2 to Aux3 66:Move disk 1 from Aux4 to Aux1 67:Move disk 1 from Aux1 to Aux2 68:Move disk 2 from Aux3 to Aux4 69:Move disk 1 from Aux2 to Aux3 70:Move disk 1 from Aux3 to Aux4 71:Move disk 3 from Aux2 to Aux3 72:Move disk 1 from Aux4 to Aux1 73:Move disk 1 from Aux1 to Aux2 74:Move disk 2 from Aux4 to Aux1 75:Move disk 1 from Aux2 to Aux3 76:Move disk 1 from Aux3 to Aux4 77:Move disk 2 from Aux1 to Aux2 78:Move disk 1 from Aux4 to Aux1 79:Move disk 1 from Aux1 to Aux2 80:Move disk 3 from Aux3 to Aux4 81:Move disk 1 from Aux2 to Aux3 82:Move disk 1 from Aux3 to Aux4 83:Move disk 2 from Aux2 to Aux3 84:Move disk 1 from Aux4 to Aux1 85:Move disk 1 from Aux1 to Aux2 86:Move disk 2 from Aux3 to Aux4 87:Move disk 1 from Aux2 to Aux3 88:Move disk 1 from Aux3 to Aux4 89:Move disk 5 from Start to Aux1 90:Move disk 5 from Aux1 to Aux2 91:Move disk 5 from Aux2 to Aux3	to Aux2 62:Move disk 4 from Aux3 to Aux4 63:Move disk 1 from Aux2 to Aux3 64:Move disk 1 from Aux3 to Aux4 65:Move disk 2 from Aux2 to Aux3 66:Move disk 1 from Aux4 to Aux1 67:Move disk 1 from Aux1 to Aux2 68:Move disk 2 from Aux3 to Aux4 69:Move disk 1 from Aux2 to Aux3 70:Move disk 1 from Aux3 to Aux4 71:Move disk 3 from Aux2 to Aux3 72:Move disk 1 from Aux4 to Aux1 73:Move disk 1 from Aux1 to Aux2 74:Move disk 2 from Aux4 to Aux1 75:Move disk 1 from Aux2 to Aux3 76:Move disk 1 from Aux3 to Aux4 77:Move disk 2 from Aux1 to Aux2 78:Move disk 1 from Aux4 to Aux1 79:Move disk 1 from Aux1 to Aux2 80:Move disk 3 from Aux3 to Aux4 81:Move disk 1 from Aux2 to Aux3 82:Move disk 1 from Aux3 to Aux4 83:Move disk 2 from Aux2 to Aux3 84:Move disk 1 from Aux4 to Aux1 85:Move disk 1 from Aux1 to Aux2 86:Move disk 2 from Aux3 to Aux4 87:Move disk 1 from Aux2 to Aux3 88:Move disk 1 from Aux3 to Aux4 89:Move disk 5 from Start to Aux1 90:Move disk 5 from Aux1 to Aux2 91:Move disk 5 from Aux2 to Aux3		

92:Move disk 1 from Aux4 to Aux1 93:Move disk 1 from Aux1 to Aux2 94:Move disk 2 from Aux4 to Aux1 95:Move disk 1 from Aux2 to Aux3 96:Move disk 1 from Aux3 to Aux4 97:Move disk 2 from Aux1 to Aux2 98:Move disk 1 from Aux4 to Aux1 99:Move disk 1 from Aux1 to Aux2 100:Move disk 3 from Aux4 to Aux1 **LAST 100** 13057:Move disk 3 from Aux3 to Aux4 13058:Move disk 1 from Aux2 to Aux3 13059:Move disk 1 from Aux3 to Aux4 13060:Move disk 2 from Aux2 to Aux3 13061:Move disk 1 from Aux4 to Aux1 13062:Move disk 1 from Aux1 to Aux2 13063:Move disk 2 from Aux3 to Aux4 13064:Move disk 1 from Aux2 to Aux3 13065:Move disk 1 from Aux3 to Aux4 13066:Move disk 5 from Aux1 to Aux2 13067:Move disk 5 from Aux2 to Aux3 13068:Move disk 5 from Aux3 to Dest 13069:Move disk 1 from Aux4 to Aux1 13070:Move disk 1 from Aux1 to Aux2 13071:Move disk 2 from Aux4 to Aux1 13072:Move disk 1 from Aux2 to Aux3 13073:Move disk 1 from Aux3 to Aux4 13074:Move disk 2 from Aux1 to Aux2 13075:Move disk 1 from Aux4 to Aux1 13076:Move disk 1 from Aux1 to Aux2 13077:Move disk 3 from Aux4 to Aux1	92:Move disk 1 from Aux4 to Aux1 93:Move disk 1 from Aux1 to Aux2 94:Move disk 2 from Aux4 to Aux1 95:Move disk 1 from Aux2 to Aux3 96:Move disk 1 from Aux3 to Aux4 97:Move disk 2 from Aux1 to Aux2 98:Move disk 1 from Aux4 to Aux1 99:Move disk 1 from Aux1 to Aux2 100:Move disk 3 from Aux4 to Aux1 **LAST 100** 39305:Move disk 3 from Aux3 to Aux4 39306:Move disk 1 from Aux2 to Aux3 39307:Move disk 1 from Aux3 to Aux4 39308:Move disk 2 from Aux2 to Aux3 39309:Move disk 1 from Aux4 to Aux1 39310:Move disk 1 from Aux1 to Aux2 39311:Move disk 2 from Aux3 to Aux4 39312:Move disk 1 from Aux2 to Aux3 39313:Move disk 1 from Aux3 to Aux4 39314:Move disk 5 from Aux1 to Aux2 39315:Move disk 5 from Aux2 to Aux3 39316:Move disk 5 from Aux3 to Dest 39317:Move disk 1 from Aux4 to Aux1 39318:Move disk 1 from Aux1 to Aux2 39319:Move disk 2 from Aux4 to Aux1 39320:Move disk 1 from Aux2 to Aux3 39321:Move disk 1 from Aux3 to Aux4 39322:Move disk 2 from Aux1 to Aux2 39323:Move disk 1 from Aux4 to Aux1 39324:Move disk 1 from Aux1 to Aux2 39325:Move disk 3 from Aux4 to Aux1		

13078:Move disk 1 from Aux2 to Aux3 13079:Move disk 1 from Aux3 to Aux4 13080:Move disk 2 from Aux2 to Aux3 13081:Move disk 1 from Aux4 to Aux1 13082:Move disk 1 from Aux1 to Aux2 13083:Move disk 2 from Aux3 to Aux4 13084:Move disk 1 from Aux2 to Aux3 13085:Move disk 1 from Aux3 to Aux4 13086:Move disk 3 from Aux1 to Aux2 13087:Move disk 1 from Aux4 to Aux1 13088:Move disk 1 from Aux1 to Aux2 13089:Move disk 2 from Aux4 to Aux1 13090:Move disk 1 from Aux2 to Aux3 13091:Move disk 1 from Aux3 to Aux4 13092:Move disk 2 from Aux1 to Aux2 13093:Move disk 1 from Aux4 to Aux1 13094:Move disk 1 from Aux1 to Aux2 13095:Move disk 4 from Aux4 to Aux1 13096:Move disk 1 from Aux2 to Aux3 13097:Move disk 1 from Aux3 to Aux4 13098:Move disk 2 from Aux2 to Aux3 13099:Move disk 1 from Aux4 to Aux1 13100:Move disk 1 from Aux1 to Aux2 13101:Move disk 2 from Aux3 to Aux4 13102:Move disk 1 from Aux2 to Aux3 13103:Move disk 1 from Aux3 to Aux4 13104:Move disk 3 from Aux2 to Aux3 13105:Move disk 1 from Aux4 to Aux1 13106:Move disk 1 from Aux1 to Aux2 13107:Move disk 2 from Aux4 to Aux1 13108:Move disk 1 from	39326:Move disk 1 from Aux2 to Aux3 39327:Move disk 1 from Aux3 to Aux4 39328:Move disk 2 from Aux2 to Aux3 39329:Move disk 1 from Aux4 to Aux1 39330:Move disk 1 from Aux1 to Aux2 39331:Move disk 2 from Aux3 to Aux4 39332:Move disk 1 from Aux2 to Aux3 39333:Move disk 1 from Aux3 to Aux4 39334:Move disk 3 from Aux1 to Aux2 39335:Move disk 1 from Aux4 to Aux1 39336:Move disk 1 from Aux1 to Aux2 39337:Move disk 2 from Aux4 to Aux1 39338:Move disk 1 from Aux2 to Aux3 39339:Move disk 1 from Aux3 to Aux4 39340:Move disk 2 from Aux1 to Aux2 39341:Move disk 1 from Aux4 to Aux1 39342:Move disk 1 from Aux1 to Aux2 39343:Move disk 4 from Aux4 to Aux1 39344:Move disk 1 from Aux2 to Aux3 39345:Move disk 1 from Aux3 to Aux4 39346:Move disk 2 from Aux2 to Aux3 39347:Move disk 1 from Aux4 to Aux1 39348:Move disk 1 from Aux1 to Aux2 39349:Move disk 2 from Aux3 to Aux4 39350:Move disk 1 from Aux2 to Aux3 39351:Move disk 1 from Aux3 to Aux4 39352:Move disk 3 from Aux2 to Aux3 39353:Move disk 1 from Aux4 to Aux1 39354:Move disk 1 from Aux1 to Aux2 39355:Move disk 2 from Aux4 to Aux1 39356:Move disk 1 from		

Aux2 to Aux3 13109:Move disk 1 from Aux3 to Aux4 13110:Move disk 2 from Aux1 to Aux2 13111:Move disk 1 from Aux4 to Aux1 13112:Move disk 1 from Aux1 to Aux2 13113:Move disk 3 from Aux3 to Aux4 13114:Move disk 1 from Aux2 to Aux3 13115:Move disk 1 from Aux3 to Aux4 13116:Move disk 2 from Aux2 to Aux3 13117:Move disk 1 from Aux4 to Aux1 13118:Move disk 1 from Aux1 to Aux2 13119:Move disk 2 from Aux3 to Aux4 13120:Move disk 1 from Aux2 to Aux3 13121:Move disk 1 from Aux3 to Aux4 13122:Move disk 4 from Aux1 to Aux2 13123:Move disk 4 from Aux2 to Aux3 13124:Move disk 4 from Aux3 to Dest 13125:Move disk 1 from Aux4 to Aux1 13126:Move disk 1 from Aux1 to Aux2 13127:Move disk 2 from Aux4 to Aux1 13128:Move disk 1 from Aux2 to Aux3 13129:Move disk 1 from Aux3 to Aux4 13130:Move disk 2 from Aux1 to Aux2 13131:Move disk 1 from Aux4 to Aux1 13132:Move disk 1 from Aux1 to Aux2 13133:Move disk 3 from Aux4 to Aux1 13134:Move disk 1 from Aux2 to Aux3 13135:Move disk 1 from Aux3 to Aux4 13136:Move disk 2 from Aux2 to Aux3 13137:Move disk 1 from Aux4 to Aux1 13138:Move disk 1 from Aux1 to Aux2	Aux2 to Aux3 39357:Move disk 1 from Aux3 to Aux4 39358:Move disk 2 from Aux1 to Aux2 39359:Move disk 1 from Aux4 to Aux1 39360:Move disk 1 from Aux1 to Aux2 39361:Move disk 3 from Aux3 to Aux4 39362:Move disk 1 from Aux2 to Aux3 39363:Move disk 1 from Aux3 to Aux4 39364:Move disk 2 from Aux2 to Aux3 39365:Move disk 1 from Aux4 to Aux1 39366:Move disk 1 from Aux1 to Aux2 39367:Move disk 2 from Aux3 to Aux4 39368:Move disk 1 from Aux2 to Aux3 39369:Move disk 1 from Aux3 to Aux4 39370:Move disk 4 from Aux1 to Aux2 39371:Move disk 4 from Aux2 to Aux3 39372:Move disk 4 from Aux3 to Dest 39373:Move disk 1 from Aux4 to Aux1 39374:Move disk 1 from Aux1 to Aux2 39375:Move disk 2 from Aux4 to Aux1 39376:Move disk 1 from Aux2 to Aux3 39377:Move disk 1 from Aux3 to Aux4 39378:Move disk 2 from Aux1 to Aux2 39379:Move disk 1 from Aux4 to Aux1 39380:Move disk 1 from Aux1 to Aux2 39381:Move disk 3 from Aux4 to Aux1 39382:Move disk 1 from Aux2 to Aux3 39383:Move disk 1 from Aux3 to Aux4 39384:Move disk 2 from Aux2 to Aux3 39385:Move disk 1 from Aux4 to Aux1 39386:Move disk 1 from Aux1 to Aux2		

13139:Move disk 2 from Aux3 to Aux4 13140:Move disk 1 from Aux2 to Aux3 13141:Move disk 1 from Aux3 to Aux4 13142:Move disk 3 from Aux1 to Aux2 13143:Move disk 3 from Aux2 to Aux3 13144:Move disk 3 from Aux3 to Dest 13145:Move disk 1 from Aux4 to Aux1 13146:Move disk 1 from Aux1 to Aux2 13147:Move disk 2 from Aux4 to Aux1 13148:Move disk 1 from Aux2 to Aux3 13149:Move disk 1 from Aux3 to Aux4 13150:Move disk 2 from Aux1 to Aux2 13151:Move disk 2 from Aux2 to Aux3 13152:Move disk 2 from Aux3 to Dest 13153:Move disk 1 from Aux4 to Aux1 13154:Move disk 1 from Aux1 to Aux2 13155:Move disk 1 from Aux2 to Aux3 13156:Move disk 1 from Aux3 to Dest	39387:Move disk 2 from Aux3 to Aux4 39388:Move disk 1 from Aux2 to Aux3 39389:Move disk 1 from Aux3 to Aux4 39390:Move disk 3 from Aux1 to Aux2 39391:Move disk 3 from Aux2 to Aux3 39392:Move disk 3 from Aux3 to Dest 39393:Move disk 1 from Aux4 to Aux1 39394:Move disk 1 from Aux1 to Aux2 39395:Move disk 2 from Aux4 to Aux1 39396:Move disk 1 from Aux2 to Aux3 39397:Move disk 1 from Aux3 to Aux4 39398:Move disk 2 from Aux1 to Aux2 39399:Move disk 2 from Aux2 to Aux3 39400:Move disk 2 from Aux3 to Dest 39401:Move disk 1 from Aux4 to Aux1 39402:Move disk 1 from Aux1 to Aux2 39403:Move disk 1 from Aux2 to Aux3 39404:Move disk 1 from Aux3 to Dest		

Question 2.

With this, we have to automatically assume that the computer is using a “dirty-bit” system, meaning we do not write back to disk. With that in mind:

Row Major:

In the context of row-major order, both A and B arrays necessitate 4 pages each to store a single row due to their 4000 elements, fitting within the 2000-page active memory set. The entirety of the operation for these arrays would require a transfer of 16,000 pages each for reading and writing. Conversely, the C array is accessed column-wise, owing to the C[N-I+1,J] indexing. Each element access potentially incurs a new page load, totaling 16,000 page reads. Writes to A and B mirror the read pattern, culminating in an aggregate of 16,000 page writes.
32,000 page reads (16,000 each for A/B and 16,000 for C)
32,000 page writes (16,000 each for A and B)

Column-Major:

In a column-major configuration, every 1000 elements of A and B arrays in each column potentially demand a fresh page load into the memory, echoing the 16,000 pages for both reading and writing operations. The C array, influenced by its unique indexing, is accessed row-wise, aligning its behavior with A and B in the row-major scenario and resulting in 16,000 page reads. The writing operations for A and B, constrained by the column-major orientation, also accumulate to 16,000 page writes.
48,000 page reads (16,000 each for A, B, and C)
32,000 page writes (16,000 each for A and B)

Question 3.

In essence, for optimal performance, we should ensure the second element is always the median, yielding balanced partitions. For the slowest performance, the assignment should be such that the second element is the smallest or largest, leading to highly unbalanced partitions and increased recursive calls, significantly slowing down the execution.

Optimal Execution:

When QuickSort utilizes the last element as the pivot, optimal execution occurs when this pivot element is consistently the median of the array. In such cases, the array is consistently divided into two approximately equal parts, which is the ideal scenario for QuickSort. As the array size n approaches infinity, if $x = n/2$, the partitioning remains balanced, enabling the algorithm to achieve its best-case time complexity of $O(n \log n)$.

Worst-Case Execution:

Worst-case scenarios for QuickSort, particularly when the last element is the pivot, are typically observed under the following conditions:

Already Sorted Array: When the array is already sorted in ascending order, each pivot (being the last element) is the largest in the subarray, resulting in unbalanced partitions. The algorithm

degenerates into making recursive calls on nearly the entire array, leading to a time complexity of O(n^2).
Reverse Sorted Array: Similarly, if the array is sorted in descending order, each pivot is the smallest element in the subarray. This again results in highly unbalanced partitions and a quadratic time complexity.
Array with Identical Elements: An array with all identical elements also leads to unbalanced partitions since all elements are equal to the pivot. This skewed partitioning significantly reduces the efficiency of QuickSort.

Question 4.

```
    n = 100
[Version 1] Time used: 0 milliseconds.
[Version 2] Time used: 0 milliseconds.
    n = 200
[Version 1] Time used: 0 milliseconds.
[Version 2] Time used: 0 milliseconds.
    n = 400
[Version 1] Time used: 0 milliseconds.
[Version 2] Time used: 0 milliseconds.
    n = 800
[Version 1] Time used: 1 milliseconds.
[Version 2] Time used: 4 milliseconds.
    n = 1600
[Version 1] Time used: 7 milliseconds.
[Version 2] Time used: 47 milliseconds.
    n = 3200
[Version 1] Time used: 41 milliseconds.
[Version 2] Time used: 193 milliseconds.
    n = 6400
[Version 1] Time used: 131 milliseconds.
[Version 2] Time used: 751 milliseconds.
    n = 12800
[Version 1] Time used: 505 milliseconds.
[Version 2] Time used: 3759 milliseconds.

[Done] exited with code=0 in 7.156 seconds
```

Let it be known that both arrays were allocated and deallocated, while also cleaning each iteration during its loop of n.

What is shown in the results is that instead of having the times be nearly identical, version 2 is actually taking an exponentially longer amount of time for each entry. Between row and column addition, you can get extremely different efficiencies running the same sequence. For smaller sequences, having either Version 1 or Version2 can prove to have no difference; however, once reaching larger sequences, the time it takes for each program to run shows significantly greater concurrence with Version 1 rather than Version 2. This is due to how each version runs.

In Version 1, matrix addition is executed by iterating through the matrices row by row. It sequentially accesses elements stored in contiguous memory locations. This alignment with the memory's structural organization fosters enhanced data locality and optimizes cache utilization.

Each cache line fetched is fully utilized before moving to the next, minimizing cache misses and thereby optimizing execution speed.

Conversely, Version 2 navigates the matrices column by column. Given that matrices are typically stored in row-major order in memory, this approach necessitates accessing elements that are not stored contiguously in memory. The resultant frequent cache misses and the need to constantly load new cache lines lead to an escalation in execution time, especially pronounced with the augmentation of matrix sizes.

In light of these findings, while theoretical models provide foundational insights, their integration with practical considerations associated with system architecture and memory configurations is indispensable. It fosters a more nuanced, comprehensive understanding, enabling the development of optimized, efficient computational solutions, particularly pertinent in applications involving extensive matrix operations and large datasets.

C++

```
#include <iostream>
#include <chrono>

void printTimeDiff(std::chrono::steady_clock::time_point start, int ver)
{
std::cout << "[Version " << ver << "] Time used: "
<<
std::chrono::duration_cast<std::chrono::milliseconds>(std::chrono::steady_cloc
k::now()
- start).count()
<< " milliseconds.\n";
}

int main(int argc, char *argv[])
{
int ns[10] = {100, 200, 400, 800, 1600, 3200, 6400, 12800, 25600, 51200};
int sequences = 8;

for (int seq = 0; seq < sequences; seq++)
{
int n = ns[seq];
std::cout << "\tn = " << n << '\n';

int **a = new int *[n];
int **b = new int *[n];
int **c = new int *[n];

for (int i = 0; i < n; i++)
{
a[i] = new int[n](); // Initialize to 0
```

```
b[i] = new int[n]();
c[i] = new int[n]();
}

// version 1
auto start = std::chrono::steady_clock::now();
for (int i = 1; i < n; i++)
{
for (int j = 1; j < n; j++)
{
c[i][j] = a[i][j] + b[i][j];
}
}
printTimeDiff(start, 1);

// version 2
auto start_two = std::chrono::steady_clock::now();
for (int j = 1; j < n; j++)
{
for (int i = 1; i < n; i++)
{
c[i][j] = a[i][j] + b[i][j];
}
}
printTimeDiff(start_two, 2);

//deallocate
for (int i = 0; i < n; i++)
{
delete[] a[i];
delete[] b[i];
delete[] c[i];
}

delete[] a;
delete[] b;
delete[] c;
}

return 0;
}
```

Question 5. Using C++:

m=700000

```
Time taken for 3m arrays of 1MB: 4824072 microseconds
Time taken to deallocate even-numbered arrays: 320589 microseconds
Time taken for m arrays of 1.25MB: 2433279 microseconds
```

The memory management and allocation dynamics in a system are intricate, and our observations lend credence to this complexity. The binary allocation process seems to implement some defragmentation procedures during the allocation of the primary array. The consequential allocation timing showcases a noteworthy variance when juxtaposed with the initial allocation, suggesting a non-trivial overhead linked with potential defragmentation.

Experiments with the system revealed its limitations. While one would expect standard dynamic allocation techniques to ameliorate memory management, the program frequently encountered issues, especially when resorting to the traditional malloc(). This was accentuated when a 64-bit binary compilation was attempted. The endeavor to utilize the entirety of the system memory often culminated in system-wide crashes. A meticulous trial-and-error approach led to the conclusion that an m value of around 700,000 was the ceiling to which the system could be safely pushed without invoking system instability.

An intriguing observation was the relatively comparable performance of delete[] and free() methods, exhibiting analogous results within a margin of error. The time metrics also presented interesting insights. For instance, an array thrice the size of the original could be hypothetically extrapolated to occupy about 31.5 seconds, based on the measurements.

Post-deallocation, a significant portion, roughly half, of the memory from the primary sequence remained vacant. This available space becomes critical, considering the substantial memory requirements of the allocations. The primary allocation undeniably demands a substantial continuous memory chunk. The evident delay could be attributed to the inherent need for defragmentation, ensuring contiguous memory availability.

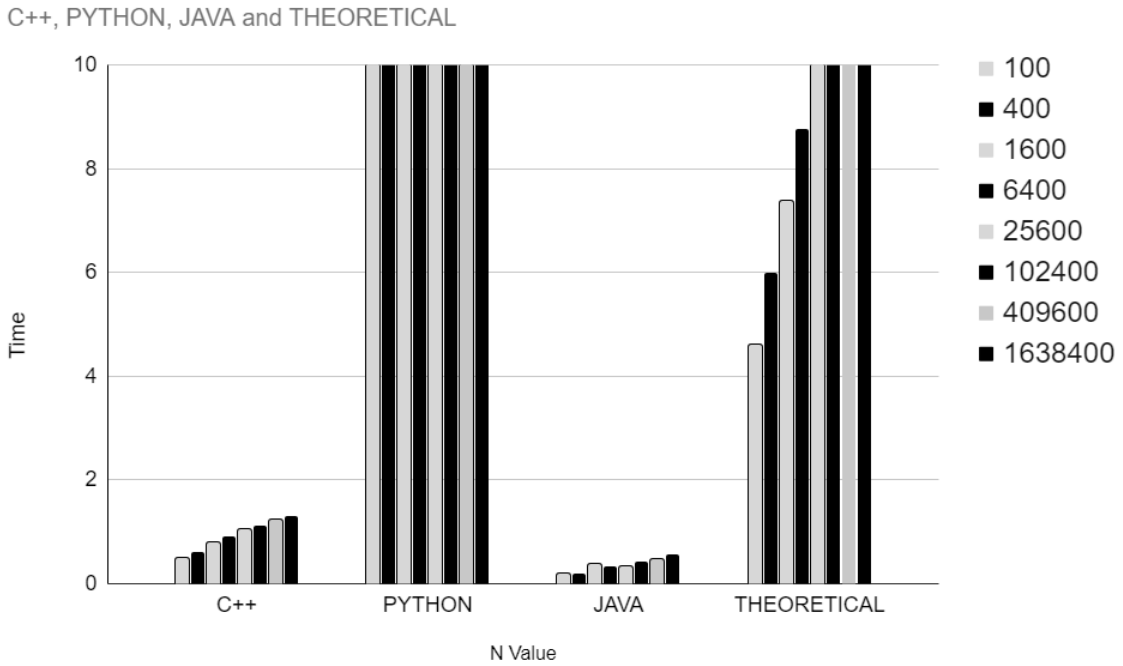
Initially, one might speculate that the second sequence's allocation would be a more time-intensive procedure, especially given the scattered nature of memory after deallocation. However, the results indicated the contrary. The memory appears to efficiently adapt, and the allocation seems less hindered by the gaps left by the previous deallocations. This might hint at the system's adeptness at finding and utilizing these proximal memory partitions.

```
#include <iostream>
#include <chrono>
#include <vector>
constexpr size_t M = 700000;
constexpr size_t MB = 1024 * 1024;
constexpr size_t ARRAY_SIZE_1MB = MB / sizeof(int);
constexpr size_t ARRAY_SIZE_1_25MB = (size_t)(1.25 * MB) / sizeof(int);
int main() {
std::vector<int*> arrays(3 * M);
auto start = std::chrono::steady_clock::now();
for (size_t i = 0; i < 3 * M; i++) {
arrays[i] = new int[ARRAY_SIZE_1MB];
}
auto end = std::chrono::steady_clock::now();
std::cout << "Time taken for 3m arrays of 1MB: "
<< std::chrono::duration_cast<std::chrono::microseconds>(end - start).count()
<< " microseconds" << std::endl;
start = std::chrono::steady_clock::now();
for (size_t i = 0; i < 3 * M; i += 2) {
delete[] arrays[i];
arrays[i] = nullptr;
}
end = std::chrono::steady_clock::now();
std::cout << "Time taken to deallocate even-numbered arrays: "
<< std::chrono::duration_cast<std::chrono::microseconds>(end - start).count()
<< " microseconds" << std::endl;
std::vector<int*> arrays2(M);
start = std::chrono::steady_clock::now();
for (size_t i = 0; i < M; i++) {
arrays2[i] = new int[ARRAY_SIZE_1_25MB];
}
end = std::chrono::steady_clock::now();
std::cout << "Time taken for m arrays of 1.25MB: "
<< std::chrono::duration_cast<std::chrono::microseconds>(end - start).count()
<< " microseconds" << std::endl;
for (size_t i = 0; i < 3 * M; i++) {
if (arrays[i]) {
delete[] arrays[i];
}
}
for (size_t i = 0; i < M; i++) {
delete[] arrays2[i];}return 0;}
```

Question 6.

Let it be known that all of these languages were run through Linux Ubuntu with small RAM and Little CPU usage. I purposely did this because I personally wanted to see how well C++ and Java could run against python; having small computer power only emphasizes the times and not much else. It really does help to show what could happen with even larger and more complex codes. General predictions of this were that C++ would be the fastest, Java second, and python having been too slow to compare. These are the results:

Amount	C++	Python	Java	Theoretical
100	.490182	16.1771	.1855	4.6015
400	.616835	37.8618	.1873	5.99146
1600	.791896	64.0281	.3795	7.37776
6400	.903031	94.5703	.3257	8.76405
25600	1.04234	130.1673	.3365	10.1503
102400	1.11827	171.0714	.4256	11.5366
409600	1.23509	216.3711	.4662	12.9229
1638400	1.31585	267.0454	.5632	14.3092



In seconds

C++

C++ is a really good coding software for things like this. It's very consistent with time complexity and keeps things low, making it ideal for programs and softwares that need to be fast. With this knowledge, we could assume pretty quickly that this will outbeat our theoretical times, considering that C++ uses such a binary structure.

Python

Only thing to summarize here is that python is quite a suboptimal language when it comes to running things quickly. This code base is not meant for things like this, and is better off doing small tasks rather than running codes that need algorithms. The times for it are way off the theoretical, which means that it's probably not worth running in this sense. Better to keep python for smaller programs rather than things that need efficiency, no offense to python users.

Java

Surprisingly, Java has better timings than everything else. Reason being is that java is more or less a middle ground rather than the superior. This is more than likely due to my IDE being more optimized for java rather than C.

C++

```
#include <iostream>
#include "time.h"
using namespace std;

void p_diff(clock_t start, int iter_len) { cout << iter_len << "\t\t" <<
float(clock() - start) / CLOCKS_PER_SEC << " seconds.\n"; }

int search(int *arr, int begin, int end, int target)
{
if (begin > end)
return -1;

int mid = (begin + end) / 2;

if (target == arr[mid])
return mid;
return target < arr[mid] ? search(arr, begin, mid - 1, target) : search(arr,
mid + 1, end, target);
}
```

```
void battery(int length, clock_t start = clock())
{
int *arr = new int[length](); // initialize to 0
for (int i = 0; i < 20000000; i++)
{
search(arr, 0, length - 1, length);
}
p_diff(start, length);
}

int main(int argc, char *argv[])
{
int lengths[] = {100, 400, 1600, 6400, 25600, 102400, 409600, 1638400};
for (const int length : lengths)
{
battery(length);
}
return 0;
}
```

Python

```
import time

def p_diff(start, iter_len):
print(f"{iter_len}\t\t{time.process_time() - start:.4f} seconds.")

def search(arr, begin, end, target):
if begin > end:
return -1

mid = (begin + end) // 2

if target == arr[mid]:
return mid

return search(arr, begin, mid - 1, target) if target < arr[mid] else
search(arr, mid + 1, end, target)

def battery(length, start=time.process_time()):
arr = [0] * length # initialize to 0
for i in range(20000000):
search(arr, 0, length - 1, length)
```

```
p_diff(start, length)

def main():
lengths = [100, 400, 1600, 6400, 25600, 102400, 409600, 1638400]
for length in lengths:
battery(length)

if __name__ == "__main__":
main()
```

Java

```
public class Timings{
public static void main(String[] args) {
int[] lengths = {100, 400,1600,6400,25600,102400,409600,1638400};
for (int length : lengths) {
long startTime = System.nanoTime();
battery(length);
long endTime = System.nanoTime();
double elapsedTimeInSeconds = (endTime - startTime) / 1e9;
System.out.printf("%d\t\t%.4f seconds.%n", length, elapsedTimeInSeconds);
}
}

public static int search(int[] arr, int begin, int end, int target) {
if (begin > end) return -1;

int mid = (begin + end) / 2;

if (target == arr[mid]) return mid;
return (target < arr[mid]) ? search(arr, begin, mid - 1, target) : search(arr,
mid + 1, end, target);
}

public static void battery(int length) {
int[] arr = new int[length];
for (int i = 0; i < 20000000; i++) {
search(arr, 0, length - 1, length);
}
}
}
```

Question 7.

Using the input:

```
std::vector<char> chars = {'A', 'B', 'C', 'D', 'E', 'F', 'G'};
std::vector<int> amt = {500, 400, 300, 200, 100, 50, 25};
```

We get the result:

```
C : 00
G : 01000
F : 01001
E : 0101
D : 011
B : 10
A : 11

[Done] exited with code=0 in 0.594 seconds
```

This code is giving us a result that has a time complexity of O(n Log2n). To implement the optimal Huffman algorithm with a time complexity of O(n log2n), where 'n' is the number of code symbols, you need an efficient data structure to find the two elements with the smallest probability at each step. The core idea behind this algorithm is to build a Huffman tree that assigns shorter codes to more frequent symbols and longer codes to less frequent symbols, thereby achieving optimal data compression. To achieve this, you can use a priority queue with minHeap (often implemented as a binary heap) to efficiently select and merge nodes with the lowest probabilities during tree construction. In each step, you extract the two nodes with the smallest probabilities from the priority queue, create a new internal node with their combined probability, and insert it back into the queue. This process continues until you have a single root node, which represents the complete Huffman tree. By using a priority queue, you ensure that finding and merging the smallest probabilities is efficient, leading to a time complexity of O(n log2n) for constructing the Huffman tree. The resulting Huffman codes will have the property of optimal data compression, with shorter codes assigned to more frequent symbols and longer codes assigned to less frequent symbols, as shown with the results. With that, the highest number will always take the least amount of space, making this kind of implementation perfect for sending data back and forth such as SMS. That's where this implementation of low space, high numbers can really be shown to be more efficient.

```
#include <iostream>
#include <vector>
#include <queue>

class HuffmanNode
{
public:
char CH;
unsigned freq;
HuffmanNode *left;
HuffmanNode *right;

HuffmanNode(char CH, unsigned freq)
{
this->CH = CH;
this->freq = freq;
left = right = nullptr;
}
};

struct NodeComparator
{
bool operator()(HuffmanNode *left, HuffmanNode *right)
{
return left->freq > right->freq;
}
};

void printHuffmanCodes(HuffmanNode *root, std::string code = "")
{
if (!root)
return;

if (root->CH != '$')
std::cout << root->CH << " : " << code << "\n";

printHuffmanCodes(root->left, code + "0");
printHuffmanCodes(root->right, code + "1");
}
```

```
void buildHuffmanTree(std::vector<char> chars, std::vector<int> amt, int size)
{
std::priority_queue<HuffmanNode *, std::vector<HuffmanNode *>, NodeComparator>
minHeap;

for (int i = 0; i < size; ++i)
minHeap.push(new HuffmanNode(chars[i], amt[i]));

while (minHeap.size() != 1)
{
HuffmanNode *left = minHeap.top();
minHeap.pop();

HuffmanNode *right = minHeap.top();
minHeap.pop();

HuffmanNode *internalNode = new HuffmanNode('$', left->freq + right->freq);

internalNode->left = left;
internalNode->right = right;

minHeap.push(internalNode);
}

printHuffmanCodes(minHeap.top());
}

int main()
{
std::vector<char> chars = {'A', 'B', 'C', 'D', 'E', 'F', 'G'};
std::vector<int> amt = {500, 400, 300, 200, 100, 50, 25};
int size = chars.size();
buildHuffmanTree(chars, amt, size);
return 0;
}
```