Yakup Talha Yolcu CSE321 , HW2

1/ a) $x(n) = a \cdot x\left(\frac{n}{b}\right) + f(n)$ $n = b^k$ $x(1) = c$ $a \geq 1$ $b \geq 2$ $c > 0$

$T(n) = 16T(n/4) + n!$ $a = 16$ $b = 4$

$T(n) = \Theta(n^{\log_b a})$ $f(n) = O(n^{\log_b a - \varepsilon})$

$\Theta(n^{\log_b a} \log n)$ $f(n) = \Theta(n^{\log_b a})$

$\Theta(f(n))$ $f(n) = \Omega(n^{\log_b a + \varepsilon})$

$n! \in \Omega(n^{\log_{16} 4 + \varepsilon})$ $\varepsilon = 1/4$

$\in \Omega(n^4)$ So $T(n) = \Theta(n!)$

b) $T(n) = \sqrt{2}\,T(n/4) + \log n$ $a = \sqrt{2}$ $b = 4$ $\log n \in O(n^{1/4})$

So $T(n) = \Theta(n^{1/4})$

c) $T(n) = 8T(n/2) + 4n^3$ $a = 8$ $b = 2$ $f(n) = 4n^3$ $4n^3 \in \Theta(n^3)$

So $T(n) = \Theta(n^3 \log n)$

d) $T(n) = 64T(n/8) - n^2 \log n$ $-n^2 \log n$ is negative so it can't be calculated.

e) $T(n) = 3T(n/3) + \sqrt{n}$ $a = 3$ $b = 3$ $f(n) = \sqrt{n}$ $\sqrt{n} \in O(n)$

So $T(n) = \Theta(n)$

f) $T(n) = 2^n T(n/2) - n^n$ $-n^n$, it can't be calculated

g) $T(n) = 3T(n/3) + \frac{n}{\log n}$ $n/\log n$ is not polynomial it can't be solved

2/ a) how many part $\Rightarrow a$
part size $\Rightarrow b$
cost $\Rightarrow f(n)$

$T(n) = 9T(n/3) + n^2$ $a = 9$ $b = 3$ $f(n) = n^2$

$n^2 \in \Theta(n^2)$

so $T(n) = O(n^2 \log n)$

b) $T(n) = 8T(n/2) + n^3$ $n^3 \in \Theta(n^3)$ so $T(n) = O(n^3 \log n)$

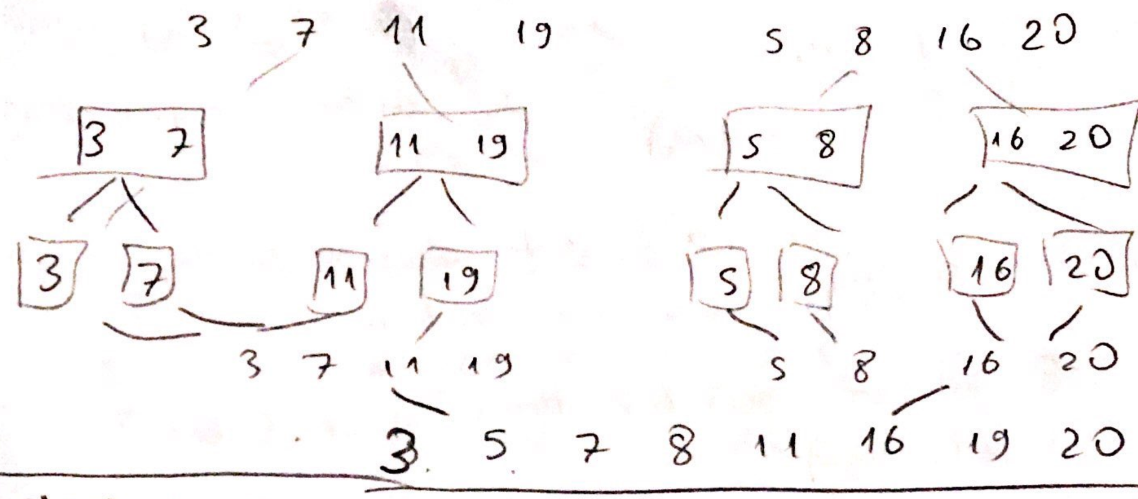
c) $T(n) = 2T(n/4) + \sqrt{n}$ $\sqrt{n} \in \Theta(\sqrt{n})$ so $T(n) = O(\sqrt{n} \log n)$

I would choose Algorithm 7 because it has less time complexity

3/ c) Merge sort Input 3 7 11 19 5 8 16 20

i) worst case: maximum number of comparisons. ⇒ Every element of the array will be compared number of comparison is n

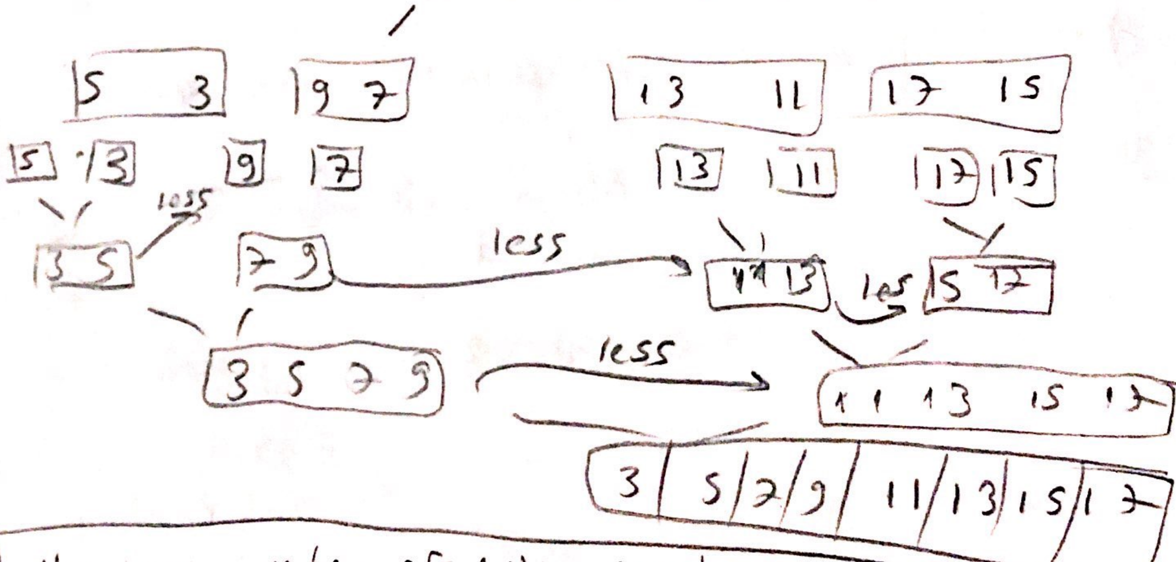
3 7 11 19 | 5 8 16 20



ii) Input: 5 3 9 7 13 11 17 15

To be able to have minimum number of comparisons we need to have maximum element of the left sublist is less than the first element of the other sublist

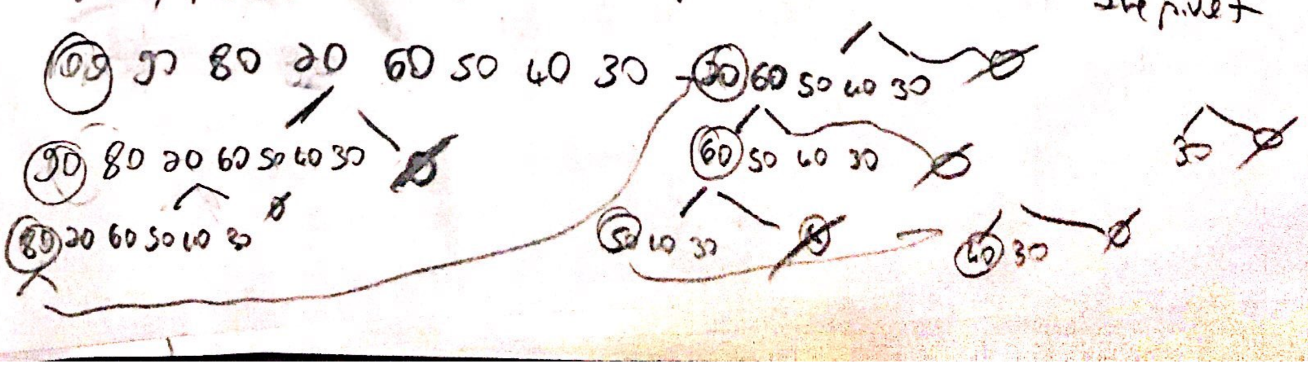
5 3 9 7 13 11 17 15



6) i) Maximum number of swap operations:

If the array is already sorted then we have maximum number of swap operations, Array could be inversely sorted it doesn't matter

If so, there is no element to compare at the side of the pivot



6) ii) Minimum number of swap operations. This case occurs when median element is pivot or partitions are as evenly balanced as possible difference between number of elements are 0 or 1

19 12 14 16 17 18 13 (15) → pivot

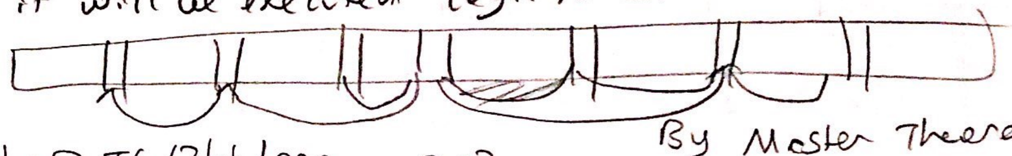
12 14 (13) 16 17 18 (19)

12 14

16 (17) 18

16 ∅

4) At each else problem is divided into 2 parts and we have n/2 size. and we have assignment which is constant. But it will be executed logn times



By Master Theorem

$T(n) = 2T(n/2) + \log n$ $a=2$

$b=2$ $\log n \in O(n)$ so

$T(n) = \Theta(n)$

5)
```
procedure Quicksort(boxes[0-n], gifts[0-n], ---high, low)
        if len(boxes)==1 then return 0;
    if low < high
            pi = partition(gifts, low, high, boxes)
            quicksort(gifts, low, pi-1, boxes)
            quicksort(gifts, pi+1, high, boxes)

    endif
end

procedure partition (gifts, low, high, boxes)
        i = (low - 1)
        pivot = boxes(high)

        for j = low to high
            if gifts[j] <= pivot
                i = i+1
                swap gifts[i] and gifts[j]
                swap boxes[j] and gifts[j]
            endif
        endfor
        swap gifts[i+1] and gifts[high]
        swap boxes[i+1] and gifts[high]
        return (i+1)
end
```

Yakup Talha Yolcu 1801042609 CSE321 HW2

Before going quicksort algorithm, we need to make sure that order of box and gift array is same. So we have order procedure

```
procedure order (boxes[0 — n], gifts[0 — n])
    temp=0
    for i=0 to len(gifts)
        if gifts[i] != boxes[i]
            for j=0 to len boxes
                if boxes[j] == gifts[j]
                    temp=j
                    break
                endif
            endfor
            swap boxes[j] and boxes[i]
        endif
end
```

Analyzing of algorithm:

Order algorithm: worst case: all elements should be changed. $\Theta(n^2)$

Best case: $\Theta(1)$

Analyzing of quicksort: Same quicksort algorithm is used.

Best case: if the list is divided into equal parts

$\Theta(n\log n)$

Worst case: $\Theta(n^2)$

Average case: $A(n) = 2(n+1) \cdot H(n+1) - 3(n+1)$

$\hookrightarrow \ln(n+1)$

$= \Theta(n\log n)$

Scanned with CamScanner