SWE Assignment

Name: Abhiroop Mukherjee
Enrlment No.: 510519109

Q1) Various Types of COCOMO

- COCOMO (constructive cost model) is a regression based model, based on number of lines of code (LoC)

→ It is a procedural cost estimate model for software projects & it is often used as a process of reliably predicting the various parameters associated with the projects like, size ,effort, cost, time & quality.

Types of COCOMO

(i) Basic COCOMO Model
(ii) Intermidiate COCOMO Model)
(iii) Detailed COCOMO Model

I) Basic COCOMO Model

→ Used for quick & slightly rough calculation of software cost.

→ Accuracy is restricted to some extent ~~data~~ due to absence of sufficient factor consideration

$$\text{Effort} = a\,(\overset{\text{kilo}}{k}\,LoC)^{b}$$
$$\text{time} = c\,(\text{effort})^{d}$$

$$\text{person\_required} = \frac{\text{effort}}{\text{time}}$$

where a,b,c,d are constants.

II Intermidiate COCOMO Model

→ No system's effort & schedules can be calculated using only LoC as metric.

→ here factors like reliability, experiance, capability, etc are taken into account.

→ These factors are known as cost drivers & intermidiate COCOMO Model utilizes 15 such drivers for cost estimation.

→ These attributes are.

- (a) Product Attributes
  - (i) → Required Software Reliability extent
  - (ii) → Size of application database
  - (iii) → Complexity of the problem
- (b) P~~ersonel~~ Hardware Attributes
  - (i) → ~~Analyst Capability~~ Runtime performance
  - (ii) → Memory Contraints
  - (iii) → The volatility of the VM Enviorment
  - (iv) → Required Turnaround time
- (c) Personal Attributes
  - (i) → Analyst Capability
  - (ii) → Software Engineering Capability
  - (iii) → Applications Experience
  - (iv) → Virtual Machine Experience
  - (v) → Programming Language Experience
- (d) Project Attributes
  - (i) → Use of Software tools
  - (ii) → Application of Software Engineering Method
  - (iii) → Required Development Schedule

→ These 15 values are taken together to calculate <u>EAF (Effort Adjustment Factor)</u>

$$\text{Effort} = a\,(KLOC)^b \times EAF$$

(III) <u>Detailed COCOMO</u> incorporates all characteristics of intermidiate COCOMO with an assesment of the cost driver's impact on each step of the software engineering process.

→ Here, the whole software is divided into different modules & then COCOMO is applied in different modules to estimate effect & sum the effort.

→ The six phases of detailed COCOMO
(i) Planning & Requirement
(ii) System Design
(iii) Detailed Design
(iv) Module code & test
(v) Integration & Test
(vi) Cost Constructive Model

→ The effort is calculated as a function of program size & a set of cost drivers.

Q2) Risk Information Sheet & Risk Table

I) Risk Information ~~Table~~ Sheet (RIS)

→ A risk information sheet is a means of captoring information abot a risk.

→ RIS are used to document new risks as they are identified.

→ They are also used to modify information about all the risks that are managed.

→ It is a form that can be submitted to the appropriate person or included in a database with other project risks.

→ In absence of database, this becomes the primary mens of documenting & retaining information about a risk.

→ The basic feature of a ~~risk~~ RIS is that it provides a standardized formate so risks information is readily accessible & understandable.

→ Components of RIS
- → risk id, date, probability, impact, description
- → refinement, mitigation / monitoring
- → management / contingency plans / trigger
- → current status
- → originator, assigned staff member.

(II) Risk Table

→ A Risk Table is a simple technique for risk projection which involves structured tabulation of numerical representation of various ~~rec~~ risks, their implications & probability, thereby providing a way to prioritize risk management.

→ Components ~~w~~/ fields in a risk table.

(i) Category :

→ The risks are categorized as per standard categorization codes (such as "PS" for "project size" risk) in order to enlist them properly.

(ii) Probability

→ The probability of occurence of each risk

→ ~~calculated by~~

(iii) Impact

→ Determined based on their ~~im~~ individal implication

→ Categorized as
- → catatrophic
- → critical
- → marginal
- → negligible.

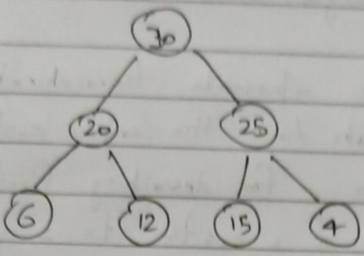
(iv) RMMM plan

Q3 Max heap creation algorithm through Desk Checking.

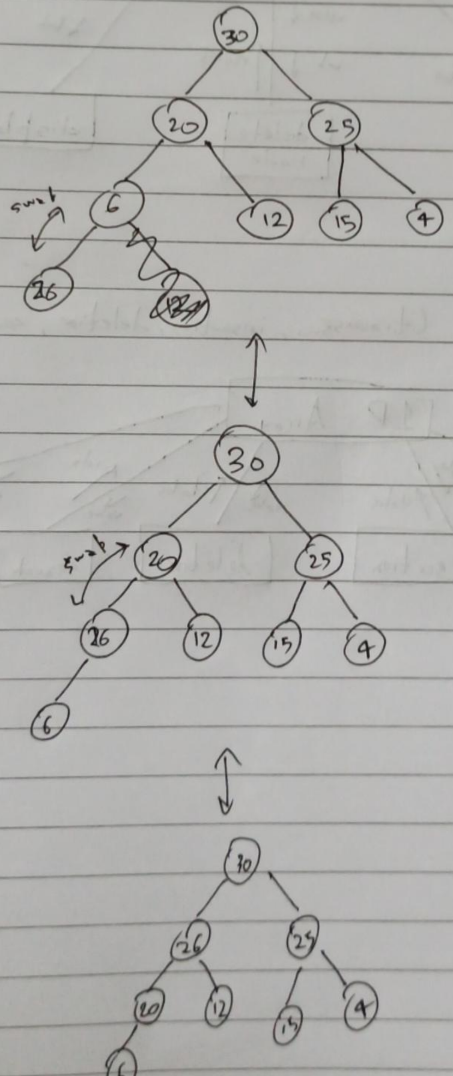
Max-healfy Pseudocode

```
                                   array    index of node
Procedure  Max-Heapify  (B , s) {
    left  = 2s           // left parent
    right = 2s + 1       // right parent

    if left ≤ B.length  && B[left] > B[s]  then
        largest = left
    else
        largest = s

    if right ≤ B.length  && B[right] > B[largest] then
        largest = right
    else
        largest = s

    if largest ≠ s then
        swap (B[s], B[largest])
        Max-Heapify (B , largest)
}
```

Eg: consider already made tree.



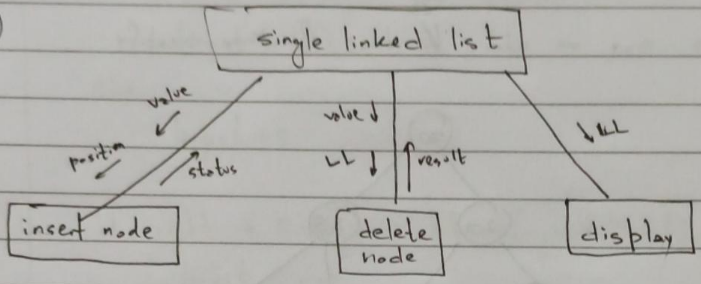
→ lets say we add $\overset{26}{16}$ here & than, heapify



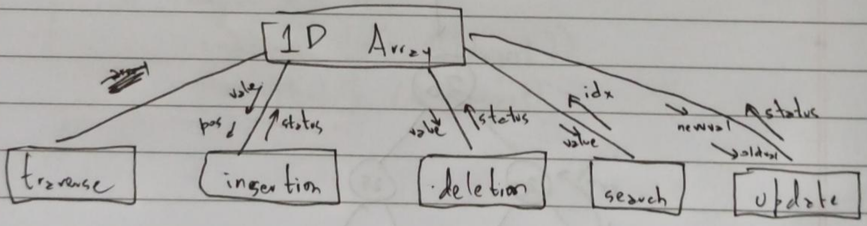
Q.) Structure Chart of basic operations (insert, delete, display) of a singly linked list & an 1D array

→ Structure Chart represents hierarchical structure of modules & breaks down the entire system into lowest functional modules, for describing functions & subfunctions of each module of a system to a greater detail

(a)



(b) 1D array (traverse., insertion, deletion, search, update)


1D Array
value
pos
status
value
status
idx
value
newval
oldval
status
traverse
insertion
deletion
search
update