1. Што е редица?

Линеарна податочна структура каде елементите се додаваат од едната страна (опашот), а се пристапуваат од другата страна (главата).

1. Кога е најдобро да се користи динамичко програмирање?

Кога потпроблемите се преклопуваат.

1. Со што најдобро би се превртеле елементите во низа?

Стек.

1. За адресирање презимиња се користи:

В-дрво.

1. Кога е графот прост?

Кога нема циклуси и паралелни ребра.

1. Дали ако нема место во CD тогаш со алчен би се добиле оптимални резултати?

Не.

1. Сложеноста на алгоритми се мери со?

Ниту еден од понудените одговори (понудени: брзина, капацитет...).

1. Кај приоритетна редица најголемиот член е?

Прв излегува.

1. Комплексноста на пристапувањето до елементи преку содржина во низа?

О(n).

1. На едно CD има N песни секоја со меморија SI. CD-то е преполно дали за бришење на песни за да се ослободи меморија е точно да се користи greedy?

Не.

1. Што е точно за дрво?

До секој јазол има различен пат почнувајќи од коренот.

1. Колкава е сложеноста за хеш табела со затворени кофички?

О(1) и О(n).

1. На кој принцип работи стекот?

First In Last Out (FILO)

1. На кој принцип работи редот?

First In First Out (FIFO)

1. Кога е НАЈНЕповолно да користиме матрица на соседство и дадени број на јазли и ребра.

Треба да се избере што поголем број на јазли и што помал број на ребра зошто матрица на

соседство е поволна кога имаме мал број на јазли и што поголем број на ребра.

1. Што е точно за B-дрвата?

Кај B-дрвата сите листови (термални јазли) се на исто ниво.

1. Комплексноста да стигнеш до елемент во неподредена низа?

О(n).

1. Даден е нетежински и неориентиран граф. Алгоритам за минимално растојание од јазол С до било кој друг јазол.
   1. пребарување во ширина (Breadth first search); 🡨
   2. внатрешен јазол;
   3. има најмалце едно дете.
2. Што функција има ОBHT со повеќе од 1 step?

Помалку кластери.

1. Комплексност за бришење на првиот елемент од листата?

О(1).

1. Комплексност за додавање елемент во низа?

О(n).

1. При еднократна ротација на јазол?

Inoreder изминување не се менува (понудени: inorder се менува, preorder се менува, preorder не се менува).

1. Матрица на соседство е најповолна ако имаме?

Најмал број на јазли и што поголем број на ребра.

1. Имплементација на stack со помош на листа?

Првиот елемент на листата е врв на стекот и стекот е неограничен при имплементација со листата.

1. Имплементација на stack со помош на низа?

Stack-от има ограничена големина која зависи од големината на низата.

1. Комплексност на пристапување на елемент според индекс во низа?

О(1).

1. За што служи циклична низа?

Кај ред се користи за надминување на комплексноста.

1. Внатрешен јазол на дрво?

Има најмалку едно дете.

1. Кој е минималниот број на јазли во бинарно дрво со висина d?

d+1

1. Што е точно кај графови:

Теме што не припаѓа на ниту еден пар ребра е изолирано теме.

1. Кога принципот на раздели и владеј станува неефикасен?

Кога проблемите се преклопуваат или не се дисјункнтни. (?)

1. Бројот на нетерминални јазли кај дрво со степен 3 е помал од бројот на темринални јазли.
2. Колкава е комплексноста на best-fit решението кај knapsack алгоритмот?

О(n^2) (може и О(n\*logn)).

1. Асимптотска горна граница на фукцијата f(n)=8n+1 НЕ е функцијата?
2. g(n)=n\*n;
3. g(n)=n\*n\*n;
4. g(n)=8n+2;
5. Ниту едно од понудените. 🡨
6. Комплексност за бришење на елемент од двострана листа ако предходно е покажан елементот?

O(1). (?)

1. Комплексност за бришење на елемент од крајот на низата?

O(n).

1. Кога се применува insertion sort?

Се применува во веќе сортирана подниза.

1. Комплексноста на алгоритмот од што НЕ зависи?
2. Комплексноста на бришење 1 елемент од листа?

O(n).

1. Кој има најголема комплексност?

O(n^2).

1. Во едно CD со m мегабајти треба да се запишат песни. Вкупните MB на песните се поголеми од m. Дали greedy алгоритмот е најефикасен во овој случај?

Не.

1. Кога се користи раздели па владеј?

Кога проблемот може да се подели на дисјунктни подпроблеми.

1. Кој алгоритам е најдобар за превртување на низа?

Stack.

1. Од дадотека се сортираат 200 имиња за 200ns со bubble sort. Колку време треба за 800 имиња?

3200ns.

O(n\*n) = O(200\*200)=200ns.  
За n1 имаме дека n1=800.

800/200=4 што значи дека факторот на зголемување на input-от ни е 4  
O(n1\*n1)=O(800\*800)=O(4\*200\*4\*200)=O(16\*200\*200) = 16\*O(n\*n)=16\*200=3200ns.

1. Големо omega е:

Долна граница, односно извршување на програмата во најдобар случај.

1. Ако треба да се дијагностицира некоја болест, дали треба да се употреби greedy алгоритам?

Не.

1. Кога се употребува bubble sort?

Bubble sort се употребува кога низата е делумно сортирана зошто тогаш комплексноста е O(n).

Во општ случај комплексноста е О(n^2).

1. Ист best и worst case имаат сортирањата:

Merge и Maximum entry sort.

1. Кој од овие алгоритми најспоро ќе се реализира?

O(n^2).

1. Првите 4 итерации биле извршени и низата е сортирана во следниов редослед 1 2 4 5 3 8 7. Со кој алгоритам се сортира?

Insertion sort (или Maximum entry).

1. Што е низа?

Последователни мемориски локации. Множество на подредени парови (индекс, вредност) при што за секое појавување на индекс, постои соодветна вредност.

1. Комплексност на hash table при пребарување, додавање или бришење во просечен случај:

O(1).

1. Greedy алгоритам за мерење на температура на ракета:

Не.

1. Во stack елементите се пристапуваат од:

Врвот.

1. Алгоритам за откривање на циклуси во граф:

Depth First Search (DFS).

се користи за наоѓање најкраток пат.

1. Која е комплексноста на:

int power(int x, int n)

{

if(n == 0) return 1;

if(n == 1) return x;

return x \* power(x,n-1);

}

Одговор: O(n) – n пати ќе се повика функцијата рекурзивно

1. Комплексност на наоѓање член во несортирана низа?

O(n).

1. Ојлеров пат е?

Пат кој го користи секое ребро од графот точно еднаш.

1. Во minheap имате 3 члена, кое од следниве е точно:
   1. Се печати сортирана низа во inorder;
   2. Се печати сортирана низа во preorder;
   3. Се печати сортирана низа во postorder;
   4. Ниту едно од наведените. 🡨
2. Кој е бројот на операции и комплексноста на алгоритмот?

int suma = 0;

for(int i = 1; i <= n; ++i)

{

suma = suma\*i + i;

}

return suma;

Одговор: f(n) = 5n+4, O(n)

1. Што е точно за приоритетна редица претставена со неподредена низа?

Додавање на елемент O(1), вадење на елемент O(n).

1. Што е точно за приоритетна редица претставена со подредена низа?

Додавање на елемент О(n), вадење на елемент О(1).

1. Што може да се моделира со графови?
2. Протеински врски;
3. Шаховска табла;
4. Уште некој пример;
5. Сите предходни; 🡨
6. Кои параметри се важни кај алгоритмите за сортирање?

Брзина и меморија.

1. Првите 4 итерации биле извршени и низата е сортирана во следниов редослед 1 2 4 5 3 8 7. Со кој алгоритам се сортира?

Insertion sort (или Maximum entry).

1. Комплексност за вадење на првиот елемент од листа?

О(1).

1. Ако QuickSort сортира 1000 елементи за 100ns, за колку време отприлика би сортирал 100 елементи?

6.7ns (понудени: 11.1ns, 15.7ns…).

QuickSort има комплексност O(n\*log(n)) со основа 2. Од тука следи дека за сортирање на n елементи потребни му се n\*log(n) чекори.

n=1000, па следи 1000\*log(1000)=100ns.

1000\*9.966=100ns.

Една инструкција се пресметува за приближно 0.01ns.  
Откако пресметавме колку време е потребно за една инструкција, можеме да го пресметаме времето за QuickSort со n=100, бидејќи овој алогритам ќе има 100\*log(100) инструкции.  
  
log(100) е 6.643 заокружено, така да за овој алгоритам ќе се потребни 6.643 \* 100 инструкции односно 664.3.  
Ако го помножиме овој број со времето по инструкција кое го пресметавме погоре, се добива 664.3 \* 0.01 ns = 6.643ns.

1. MergeSort се користи кога:

Ни се потребни загарантирани перформанси.

1. Комплексност за пристапување на последен елемент во листа?

О(n).

1. Ако сакаме да искористиме во нашата апликација абстрактен податочен тип, тогаш треба:

Да ги специфицираме множеството вредности и операциите над тие вредност.

1. Кај сортирање комплексноста на алгоритмот зависи од?

Бројот на извршени едноставни операции.

1. Која од овие е асимптотска граница на фукнцијата f(x)=x^2+3x+4?
2. x
3. x\*log(x)
4. 2x^3 🡨
5. Ниту едно од понудените
6. Комплексноста на бришење на елементи кај DLL?

О(n).

1. Како се дефинира големо сигма?
2. На едно CD има N песни со вкупна големина D. CD-то има меморија со големина М. За D>M дали со greedy алгоритам можеме да определиме на CD-то да ставиме најголем број на песни?

Да.