Greedy graph Colouring

⇒ Ensure usage of not more than 'd+1' colours where 'd' ⇒ max degree of any vertex in graph

Time Complexity : $O(V^2 + E)$

⇓

As we have to check adjacency list of the vertex 'V' for 'V' vertex to check availability of Colour

Better Algorithm

D Satur

⇓

$O(V^2)$

Welsh-Powell

$O(V^2)$

These are better algorithms in terms of no of ~~colours~~ Colours used for the given graph

Shanon Janon

↳ $T(n) = T(k) + T(n-k) + O(n)$

Worst Case

if $k = n-1$

↳ $O(n^2)$

Best Case

⇓

if $k = \frac{n}{2}$

↳ $O(n \log n)$

Min Heap Concept is used

Huffman Encoding

- ~~Heapify~~ Heapifying of n nodes => $O(n)$
- Extract Min Heap => $O(\log n) \times 2$
- Insert in Heap => $O(\log n)$
- Traverse the Tree => $O(n)$

So, Time Complexity => $O(n \log n)$ → worst

Best => $O(n)$ => (If frequencies are already sorted)

N Queen problem

$T(n) = n * T(n-1) + O(n^2)$

$T(n) = n!$

Rat in a Maze problem

$O(2^{n^2})$

Using upper bound method.

Scanned with CamScanner

m- Colouring Backtracking

$\hookrightarrow O(m^V)$

$\hookrightarrow$ If Bfs method used instead of backtracking $O(V+E)$

Hamiltonian Path

$T(N) = N * T(N-1) + O(1)$

$\hookrightarrow T(N) = N!$

Ford Fulkerson / Edmond Karp

$O(\text{maxflow} * E)$

$\hookrightarrow O(VE^2)$

Knapsack

$\hookrightarrow O(n^2)$

Coin change problem $\Rightarrow O(n^2)$

Longest common subsequence $\Rightarrow O(n^2)$

Matrix Chain Multiplication $\Rightarrow O(n^3)$

All pair shortest path

(~~[illegible]~~ $\Rightarrow O(V^3)$

(Floyd Warshall)

Longest Increasing Subsequence $\Rightarrow O(n^2)$ ~~nlogn~~

↳ But a better approach is $O(n\log n)$