

DV1: DATAVETENSKAPENS BYGGSTENAR

Analysmoment av OU7

Theodor Jonsson

2022-02-02

Inledning

Den relativa värstafallstidskomplexiteten för en bra implementation av en djupet-först traversering är $O(n + m)$, där n är antalet noder och m är antalet bågar. I denna rapport ska vi undersöka vad den absoluta tidskomplexiteten blir för algoritmen som visas i Algoritm 1. Samt hur den förhåller sig till den relativa tidskomplexiteten. Vi använder Algoritm 1 i en graf som är konstruerad som Fält av Mängd. Fältet implementeras som en array av Mängd. Mängd konstrueras som en dynamisk Bitvektor.

Algorithm depthFirst(Node n , Graph g)

Input: A node n in a graph g to be traversed

- visit(n) // Marks node n as visited
- neighbourSet = neighbours(n , g);
- for each node v in g do
 - if v is a member of neighbourSet then
 - if not visited(v) then
 - depthFirst(v , g)

Algoritm 1. en rekursiv djupet-först implementation för datatypen riktad graf implementerad med hjälp av datatypen mängd

Resultat

visit(n) byter en boolean variabel från false till true detta ger en komplexitet av $O(1)$. neighbourSet ger $O(1)$ för de är en pekare till Mängden, for each ger $O(n)$ för den loopar igenom alla noder sedan båda if-satserna inom for loopen ger $O(1)$ till sist så är detta en rekursiv funktion vilket innebär att algoritmen går igenom detta n gånger om denna graf är sammanhängande. Detta ger en sammanlagd tidskomplexitet av $O(n)(O(1) + O(1) + O(n)(O(1) + O(1)))$ vilket kan förenklas till $O(n^2)$. Tidskomplexiteten av $O(n^2)$ blir på grund av att varje nod kollas igenom en gång när den blir input till algoritmen och andra gången när den blir kollad under steget for each node v in g do.

Diskussion

Den relativa tidskomplexiteten av $O(n + m)$ stämmer in bra på den absoluta tidskomplexitet av $O(n^2)$. Om man skriver om $n + m$ alltså antal noder plus antal bågar och vi vet att max antal bågar är $n^2 - n$ så blir de $n + n^2 - n$ och båda n tar ut varandra vilket blir n^2 . Men i en graf där de är mindre bågar fallerar detta då blir $O(n + m)$ mindre än $O(n^2)$. Om algoritmen hade istället för for each node v in g do varit for each node v in neighbourSet, alltså att den kollar igenom varje neighbour till noden istället för varje nod i grafen skulle tidskomplexitet nog vara $O(n + m)$