

Greining reiknirita: Verkefni 3

Kennari: Páll Melsted

Bjarki Geir Benediktsson

Bjarni Jens Kristinsson

Tandri Gauksson

Skil: 6. apríl 2014

1 Lýsing á reikniritinu

Við byggðum lausn okkar á klasanum `KruskalMST.java`¹ sem er að finna á heimasíðu bókarinnar *Algorithms* eftir Sedgewick og Wayne. Smíður klasans tekur inn óstefnt vigtað net $G = (V, E, W)$ og notar reiknirit Kruskals til að finna minnsta spanntré mst og vigt þess w .

Við skrifuðum nýja aðferð `secondMSTweight(EdgeWeightedGraph G, Edge e)` sem að finnur vigt minnsta spanntré netsins $G_e = (V, E \setminus \{e\}, W)$. Í `keyrsluforrit.java` finnum við fyrst mst og prentum vigt þess. Síðan tökum við hvern legg $e \in mst$ (raðað í vaxandi röð eftir fyrri hniti e) og finnum minnsta spanntré netsins G_e og prentum legginn e og vigt þess w_e á staðalúttak.

Við útfærðum `secondMSTweight(EdgeWeightedGraph G, Edge e)` svipað og smíðinn fyrir `KruskalMST`. Í staðinn fyrir að fara línulega í gegn um alla leggi V og kanna hvor t þeir tengja saman samhengispætti í skóginum fyrir netið þá byggjum við upp UF hlutinn með öllum leggjum úr $E \setminus \{e\}$. Þetta skilur okkur eftir með tvo samhengispætti, eða tvö tré í skóginum, og þá förum við í vaxandi röð í gegn um E og könnum hvort að hnúturinn sé e og ef ekki hvort hann tengir saman trén tvö. Ef hann gerir það þá erum við komin með minnsta spannandi hluttré fyrir G_e .

`KruskalMST.java` notast við eftirfarandi klasa frá sömu höfundum (okkar breytingar eru taldar upp á eftir hverjum og einum klasa):

- `EdgeWeightedGraph.java`²
 - Breyttum öllum línum sem búa til nýjan hlut af taginu `Edge` til að nota heiltöluvigt í stað fleytitölu (sjá athugasemdir við `Edge.java`)
- `Queue.java`³
- `Edge.java`⁴
 - Breyttum vigtinni yfir í að vera heiltala í stað fleytitölu
 - `toString()` aðferðin gefur örlítið öðru vísi strengjaframsetningu á hnútunum
 - Bættum við `toString2()` til að prenta út hnút og vigt eins og er í `.out` skránum sem Páll gefur upp á heimasíðunni
 - Bættum við klasann `lexiCompare` sem ber saman hnúta eftir orðabókaröðun
- `MinPQ.java`⁵
- `UF.java`⁶

¹<http://algs4.cs.princeton.edu/43mst/KruskalMST.java.html>

²<http://algs4.cs.princeton.edu/43mst/EdgeWeightedGraph.java.html>

³<http://algs4.cs.princeton.edu/43mst/Queue.java.html>

⁴<http://algs4.cs.princeton.edu/43mst/Edge.java.html>

⁵<http://algs4.cs.princeton.edu/24pq/MinPQ.java.html>

⁶<http://algs4.cs.princeton.edu/15uf/UF.java.html>

- `Bag.java`⁷
- `Stack.java`⁸

2 Tímaflækja verkefnisins

Hér er G upphaflega netið sem `KruskalMST` hluturinn er smíðaður með, V fjöldi hnúta þess og E fjöldi leggja.

Að búa til hlut af taginu `KruskalMST` tekur $O(E \log V)$ tíma. Hluturinn er minnsta spanntré G og inniheldur $V - 1$ leggi. Við köllum einu sinni á `secondMSTweight` fyrir hvern legg

⁷<http://algs4.cs.princeton.edu/13stacks/Bag.java.html>

⁸<http://algs4.cs.princeton.edu/13stacks/Stack.java.html>

3 Java kóði

3.1 KruskalMST.java

3.2 keyrsluforrit.java