**TÖL301G Formleg mál og reiknanleiki**

**Heimaverkefni 10**

**Hjörvar Sigurðsson**

*Rætt var um verkefnið við Arnar Sigurðsson*

1. Við aðlögum reiknirit Dijkstra:
   1. Haldið er utan um i) lista H af hnútum sem ekki hafa verið heimsóttir, ii) lista Þ af hæstu þekktu heildarþyngd frá upphafshnút til tiltekins hnúts, og iii) lista F af sem heldur utan um það foreldri hnúts á leiðinni sem hefur hæstu þyngd á aðliggjandi legg.
   2. Reikniritið ítrar í gegnum eftirfarandi skref þangað til að allir hnútar í netinu hafa verið heimsóttir:
      1. Athugað er hvaða hnúta við komust í úr núverandi stöðu.
      2. Athugað er allar mögulegar leiðir til þeirra hnúta úr okkar núverandi stöðu, og þyngd þeirra.
      3. Farið er í þann undirhnút sem hefur aðliggjandi legg með hæstu þyngdina.
      4. Sá undirhnútur sem farið er í er fjarlægður úr H, Þ er uppfærður með þyngstu gildum, og F er uppfærður með því foreldri sem hnútur hefur í leiðinni.
   3. Skilað er leiðinni frá Landamæri til Höll Ottókars með hæstu heildarþyngd, sem og líkunum á að hann verði gómaður, en líkurnar eru fundnar með því að margfalda saman þyngdir leiðarinnar og draga útkomuna frá 1.

Tímaflækja reikniritsins:

Reikniritið er reiknirit Dijkstra, en það hefur tímaflækju O(E log V), þar sem E er fjöldi leggja, og V er fjöldi hnúta.

Inntak:

n – Heiltala sem táknar stærð borðsins | n > 4 og n % 2 = 0.

H – Listi af tvenndum sem tákna staðsetkningu hindrana.

M – Tvennd sem táknar staðsetningu marksins.

UR – Tvennd sem táknar upphafsstöðu rauða disksins.

UG – Tvennd sem táknar upphafsstöðu græna disksins.

Reiknirit:

* Gefum okkur að reitir borðsins (og því tvenndir í inntaki) séu merktir þannig að hverjum reit er gefið númer 1, 2, ... n, þar sem fyrst er merkt í gegnum dálka einna raðar, en svo næstu raðar, o.s.frv.
* Við útbúum net, N, þar sem hver hnútur netsins táknar tiltekna stöðu borðsins, hindrana og diskanna tveggja.

*Dæmi*:

Hnútur h {

Staða R = (1, 4).

Staða G = (2, 2).

Hindranir = [(3, 2), (1, 1)].

Mark = (4, 4).

Mögulegar næsta staðsetning R = [(1, 2), (4, 4), ...].

Mögulegar næsta staðsetning G = [(2, 1), (2, 4), ...].

// Möguleg næsta staðs. disks er fundin með einföldu reikniriti þar sem borið er saman stöðu disksins við ramma borðsins sem og staðsetningu hins disksins og hindrana.

}

* Netið inniheldur því (n – fjöldi hindrana)2 hnúta, þar sem til eru (n – fjöldi hindrana)2 mögulegar staðsetningar diskanna tveggja.
* Ítrað er í gegnum hvern hnút í netinu. Fyrir hvern hnút er farið í gegnum *Mögulegar næsta staðsetning R* og *Mögulegar næsta staðsetning G*, og leggur myndaður milli hnútsins og þess hnúts í netinu sem samsvarar næstu staðsetningu R og G.
* Þar með er komið net N sem inniheldur x hnúta, og leggir milli hnútanna tákna möguleikann á að færa sig milli hnútanna með því að færa annað hvort rauða eða græna diskinn.
* Að lokum er ítrað í gegnum allar þær stöður netsins þar sem annað hvort rauði eða græni diskurinn er í markinu, og hver staða sent sem inntak t, ásamt netinu N, og stöðunni s sem samsvarar upphafsstöðu þrautarinnar, í verkefnið REACHABILITY. Ef REACHABILITY skilar True, þá skilar reikniritið True. Ef ítrunin hefur klárast án þess að REACHABILITY skilaði True, þá skilar reikniritið False.

Athugið að netið er stefnt.

* Ég gef mér, sbr. verkefnalýsingu, að ég sé með mengi S sem inniheldur n jákvæðar heiltölur, heiltölu t, og að SUBSET-SUM(S, t) gefi jákvæða niðurstöðu, þ.e. að til sé hlutmengi A.
* Til þess að finna A geri ég eftirfarandi:
  + 1. Fjarlægi fyrsta stak úr S.
    2. Sendi S og t í SUBSET-SUM(S, t). Ef niðurstaðan er jákvæð, þá held ég áfram; ef niðurstaðan er neikvæð, þá set ég fyrsta stakið aftur í S og held áfram.
    3. Endurtek skref i. og ii. með stak 2, 3, ... , n.
* Þegar ég hef ítrað í gegnum öll stök mengisins og fjarlægt þau sem ekki eru nauðsynleg fyrir jákvæða útkomu úr SUBSET-SUM sbr fyrra skrefi, þá hef ég fundið hlutmengið A og því skila ég því.

Sauðakóði:

int[] S; // Inniheldur n jákvæðar heiltölur

int t; // Heiltala

for (int i = 0; i < S.length; i++) { // Ítra í gegnum stök S.

int[] temp = S.drop(i); // Fjarlægi stak i úr S.

if (SUBSET-SUM(temp, t) == True) { // Ef stakið var óþarfi þá set ég það ekki aftur í S.

S = temp;

}

}

return S; // Hér jafngildir S hlutmengi A.

1. Eftirfarandi er lýsing og sauðakóði fyrir reiknirit sem ákvarðar TVIHLUTA. Reikniritið má útfæra með löggengri Turing-vél, og reikniritið er framkvæmt á <= O(V2) | V er fjöldi hnúta í netinu G.

P inniheldur öll ákvörðunarvandamál sem löggeng Turing-vél getur leyst á polynomial tíma, en því vitum við að TVIHLUTA er í flokkinum P.

Lýsing:

Við ítrum í gegnum alla hnúta G. Fyrir hvern hnút, h, athugum við hvort hann sé merktur; ef hann er ómerktur þá merkjum við hann 1 eða 0 og merkjum alla undirhnúta (hnúta sem h hefur legg til) sem 0 eða 1 (þá tölu sem h var ekki merktur sem); ef h er merktur, þá merkjum við alla undirhnúta með þeirri tölu 0 eða 1 sem h er ekki merktur sem. Ef við lendum einhverntímann í því að undirhnútur hefur sama merki og yfirhnúturinn h, þá segjum við G ekki tvíhluta. Þegar allir hnútar hafa verið merktir förum við einu aftur einu sinni í gegnum alla hnúta og berum saman merki hnútsins og undirhnúta hans. Ef einhver undirhnútur hefur sama merki og yfirhnúturinn, þá höfnum við og segjum G ekki tvíhluta, en ef ekki þá samþykkjum við og segjum G tvíhluta.

Sauðakóði:

{

Int i = 0;

Fyrir hnútur í G:

Ef hnútur er ómerktur:

Ef i == 0:

hnútur.merki = 0;

i++;

annars ef i == 1:

hnútur.merki = 1;

i--;

hnútarNærliggjandi [] = Listi af hnútum sem hnútur hefur legg að.

Fyrir undirhnútur í hnútarNærliggjandi:

Ef undirhnútur er ómerktur:

Ef hnútur er merktur 0:

undirhnútur.merki = 1;

Annars ef hnútur er merktur 1:

undirhnútur.merki = 0

Annars ef undirhnútur er merktur:

Ef undirhnútur.merki == hnútur.merki:

Skila Ósatt;

Fyrir hnútur í G:

hnútarNærliggjandi [] = Listi af hnútum sem hnútur hefur legg að.

Fyrir undirhnútur í hnútarNærliggjandi:

Ef undirhnútur.merki == hnútur.merki:

Skila Ósatt;

Skila Satt;

}