

# Greining reiknirita vor 2023

## Heimaverkefni 6

Skila skal þessu verkefni á vefnum [Gradescope](#).

Gradescope tekur við .pdf skjölum. Frágangur á þeim skiptir máli.

Telji nemandi að mistök hafi verið gerð við yfirferð skal tilkynna slíkt á Gradescope.

Skilafrestur er til kl. 22:00 þriðjudaginn 21. febrúar. Gangi þér vel!

### 1. Litunartala (25 stig)

*Litun* á neti er merking á hnútum sem er þannig að engir aðlægir hnútar eru í sama lit (hafa sömu merkingu). *Litunartala* nets er minnsti fjöldi lita sem hægt er að lita netið með. Ákvörðun á litunartölu neta er almennt (NP)-erfitt verkefni en verkefnið er viðráðanlegt í sértílfellum.

Látum  $G = (V, E)$  vera óstefnt net sem er þannig að gráðutala (e. degree) sérhvers hnúts er í mesta lagi 2. Setjið fram reiknirit sem ákvarðar litunartölu slíks nets í tíma  $O(|V|)$ . *Ábending:* Sérhvert tvíhlutanet hefur litunartöluna 2. Er  $G$  alltaf tvíhluta? Þið getið ennfreður notfært ykkur eftirfarandi eiginleika litunartalna (setning Brook): Litunartala nets  $H$  er í mesta lagi einum hærri en hæsta gráðutala  $H$ .

### 2. Djúpleit (25 stig)

Lát  $G$  vera stefnt net (e. directed graph) og lát  $T_G$  vera djúpleitartré þess út frá einhverjum hnúti  $s$ . Látum ennfreður  $(u, v)$  vera legg í  $G$ .

- Hvernig er hægt að segja til um hvort leggurinn  $(u, v)$  sé bakleggur eða krossleggur í djúpleitartrénu  $T_G$  með því að nota aðeins gildin for (pre) og eftir (post) fyrir hnútana?
- Er hægt að nota aðeins for- og eftir-gildin til þess að gera greinarmun á trjálegg og framlegg í  $T_G$ ? Útskýrið.

### 3. Yfirfærsla (25 stig)

Leysið dæmi 13 bls. 214 í Erickson ("The infamous Mongolian puzzle-warrior Vidrach Itky Leda ...") með því að yfirfæra það á netverkefni.

## 4. networkx (25 stig)

Útfærið lausn ykkar á dæmi 3 í Python með `networkx` pakkanum (sjá sýnidæmi í `graph_example.py`). Látið forritið prenta lengd stystu lausnar ef hún er til, ásamt lausninni sjálfri, en annars tilkynna að engin lausn sé til. Prófið forritið ykkar á þrautinni sem fylgir dæminu (mynd 5.16 í Erickson). Tölurnar í þrautinni er að finna í `h6_example_puzzle.txt`. Þið getið lesið tölurnar inn í numpy fylki með

```
D = np.loadtxt('h6_example_puzzle.txt', delimiter=',', dtype=int))
```

Ef þið viljið búa til  $n \times n$  prófunartilvik fyrir forritið ykkar þá er t.d. hægt að nota

```
D = np.random.randint(1, n, (n, n))
```

Notið svo forritið ykkar til að kanna hvort þrautir í skránum `h6_test_puzzle1.txt` og `h6_test_puzzle2.txt` hafi lausn. Sýnið lausnirnar ef þær eru á annað borð til með því að sýna úttak úr forritinu.