

TÖL203G Tölvunarfræði 2

Lokapróf
Kennari: Páll Melsted

7. maí 2021
kl. 13³⁰ – 16³⁰

Nafn og kennitala

- Leyfileg hjálpargögn eru eitt A4 blað, skrifað báðu megin. Önnur skrifleg hjálpargögn eru óheimil.
 - Prófið er 110 stig, 100 stig gefa 10 í einkunn.
 - Skráið öll svör í prófið sjálft innan reita, ef svar er fyrir utan reitinn þarf að merkja það sérstaklega.
 - Athugið að rökstyðja öll svör og að kannski er ekki best leysa dæmin í vaxandi röð.
 - Kennari mætir ekki í prófstofu á meðan prófi stendur. **Komi fram galli í spurningu verður tekið tillit til þess við yfirferð prófsins.**
 - Prófið er 9 tölusettar síður. Á síðustu blaðsíðu er yfirlit yfir API fyrir klasa úr bókinni, það þarf ekki að skila þessari síðu með prófinu og ekki skrifa neitt á þá síðu sem þið viljið að farið verði yfir.
1. (35 stig) Fjölvalsspurningar. Hvert rétt svar gefur 5 stig. Aðeins er eitt rétt svar fyrir hverja spurningu. Ekki er dregið niður fyrir röng svör. **Dragið hring** utan um rétt svar.
- (a) Tölurnar 1-10 eru geymdar í tvíleitartré. Leitað er að tölunni 3 og gildi þeirra hnúta sem eru skoðaðir við leitina eru prentaðir út. Hvaða runa getur ekki verið prentuð út?
- (A) 10, 9, 8, 6, 3
(B) 1, 7, 2, 5, 4, 3
(C) 4, 10, 3
(D) 1, 2, 10, 6, 4, 3
- (b) Tölurnar 1 til 9 eru settar á stafla og teknar af með blöndu af push og pop. Hver af eftirfarandi röðum getur komið fyrir þegar talan er prentuð út í hvert skipti sem kallað er á pop.
- (A) 2 1 4 3 5 9 8 6 7
(B) 1 2 9 8 6 7 5 4 3
(C) 2 1 9 8 7 5 6 4 3
(D) 2 3 1 5 4 7 6 8 9

- (c) Fylkið a er búið til með eftirfarandi skipun

```
int[] a = new String[N];
```

Hvert af eftirfarandi er á við hérna.

- (A) Minni fyrir fylkið er tekið frá í Kös (e. Heap) þegar gildi eru sett inn í það
 - (B) Minni fyrir fylkið er tekið frá í Kös (e. Heap) þegar new skipunin er keyrð og minni fyrir strengina er tekið frá í Kös (e. Heap) um leið.
 - (C) Minni fyrir fylkið er tekið frá á Stafla (e. Stack) þegar new skipunin er keyrð og minni fyrir strengina er tekið frá í Kös (e. Heap) um leið.
 - (D) Minni fyrir fylkið er tekið frá í Kös (e. Heap)
- (d) Hvert af eftirfarandi reikniritum gæti skilað StackOverflow villu, miðað við þær útfærslur sem var farið yfir í námskeiðinu.
- (A) Mergesort þegar fylkið er raðað
 - (B) Helmingunarleit þegar stakið sem leitað er að er ekki í fylkinu
 - (C) Innsetning á gildi í tvíleitartré
 - (D) Sink aðferðin í Heapsort.
- (e) Hvert af eftirfarandi röðunaraðferðum er stöðug (e. stable) og keyrir á tíma $O(N)$ þegar inntakið er raðað í vaxandi röð.
- (A) Quicksort
 - (B) Insertionsort
 - (C) Mergesort
 - (D) Heapsort
- (f) Gefið er samanhagandi net G með n hnúta og allir hnútar hafa gráðuna d , þar sem $d \sim \sqrt{n}$ og jákvæðar lengdir w á leggjunum. Hver af eftirfarandi staðhæfingum gildir um stystu leiðar verkefni með s sem upphafshnútt.
- (A) Fyrir hvern hnút v er til nákvæmlega ein stysta leið frá s .
 - (B) Keyrslutíminn á reikniriti Dijkstra er $n^{1.5} \log(n)$ fyrir þetta net.
 - (C) BFS getur fundið stystu leið í netinu á línulegum tíma.
 - (D) Það eru ekki nægar upplýsingar til að svara þessari spurningu.
- (g) Reiknirit hefur tímaflækjuna $T(n) = O(n \log(n))$. Hver af eftirfarandi staðhæfingum er röng
- (A) Tímaflækjan er minni en n^2 fyrir öll n .
 - (B) Tímaflækjan er minni en $C \cdot n \log(n)$ fyrir eitthvert gildi á C
 - (C) Tímaflækjan er minni en n^2 fyrir n nógu stórt
 - (D) Tvöföldunarhlutfallið stefnir á 1.

2. (5 stig) Skrifðu reglulega segð sem passar við „tvíundarstrengi með a.m.k. þrjú 0, en ekki tvö 0 hlið við hlið“ og eingöngu þá strengi.

Vantar meira pláss? ☐ Svarið er að finna _____

3. (5 stig) Gefið er eftirfarandi fall tekur inn streng s og tölu N .

```
public static String f(String s, int N) {  
    String r = s;  
    for (int i = 0; i < N; i++)  
        r += s;  
    return r;  
}
```

Hver er tímaflækjan sem fall af N og lengdinni á strengnum s . Rökstyðjið svarið vandlega og sýnið útreikninga.

Vantar meira pláss? ☐ Svarið er að finna _____

4. (5 stig) Forrit var tímamælt með eftirfarandi stærðum á inntaki N .

N	tími (sekúndur)
500	0.1
1000	0.2
2000	0.9
4000	4.1
8000	15.5
16000	63.1
32000	121.9

Hvaða tímaflækju hefur forritið sem fall af N ?

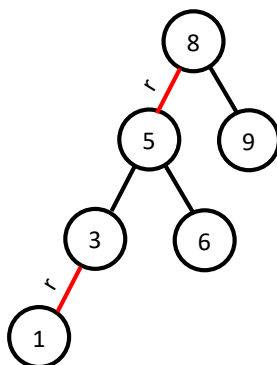
Vantar meira pláss? ☐ Svarið er að finna _____

5. (15 stig) Teiknið eftirfarandi gagnagrindur þegar búið er að setja stökin 5,3,8,1,6,9,4,7,2 inn í þessari röð.

(a) (6 stig) Lágmarkshrúgu (e. Min Heap). Hér þarf bara að teikna lokaniðurstöðu. Teiknið bæði tréð og samsvarandi fylki.

Vantar meira pláss? ☐ Svarið er að finna _____

(b) (4 stig) Vinstri hallandi rautt-svart-tvíleitartré (Left Leaning Red Black Binary Search Tree). Þetta er sú útgáfa sem við fórum yfir í námskeiðinu. Teiknið gagnagrindina eins og þegar 4 er sett inn og sýnið skrefin sem tekin eru. Táknið rauða leggi með r. Til einföldunar er tréð gefið eftir að búið er að setja inn 5,3,8,1,6,9.



Vantar meira pláss? ☐ Svarið er að finna _____

- (c) (5 stig) Hakkatafla af stærð 15 með hakkafallinu $h(k) = k^2 + 3k$ og árekstrar eru leystir með Linear Probing og taflan er aldrei stækkuð eða minnkuð. Hér þarf bara að teikna lokaniðurstöðu. (Stökin eru 5,3,8,1,6,9,4,7,2)

Vantar meira pláss? ☐ Svarið er að finna _____

6. (7 stig) Í tvíleitartre er eftirfarandi Node klasi notaður

```
private class Node {  
    private Key key;  
    private Value val;  
    private Node left, right;  
}
```

Skrifið fall findVal sem tekur inn tvíleitartre T með rót x , gildi (e. Value) p og skilar minnsta lyklinum í T sem hefur gildi jafnt og p .

Vantar meira pláss? ☐ Svarið er að finna _____

7. (8 stig) Skrifðu fall sem tekur inn stefnt net G og finnur fjölda þrennda u, v, w þ.a. (u, v) og (v, w) eru leggir í netinu en (u, w) er ekki leggur.

Vantar meira pláss? ☐ Svarið er að finna _____

8. (8 stig) Í útfærslu á Tvíleitartré (e. Binary Search Tree, BST) er notaður innri klasi eins og gefinn er upp í spurningu 6 að ofan. Útskýrið í stuttu máli af hverju við viljum að klasinn sé bæði `private` og `partur af BST klasanm.` Ef klasinn væri `public`, af hverju væri ekki góð hugmynd að `get` og `put` skiluðu tilvísun á `Node` hluti?

Vantar meira pláss? ☐ Svarið er að finna _____

9. (15 stig) Klasinn `StudentCourse` er gefinn sem heldur utan um skráningar nemenda í námskeið. Hér gerum við ráð fyrir nöfn nemenda séu ólík. Gefið er fylki a af `StudentCourse` hlutum og engir hlutir endurteknir í fylkinu.

```
public class StudentCourse implements Comparable<StudentCourse> {  
    public String name;  
    public String course;  
}
```

- (a) (4 stig) Útfærið `compareTo` aðferðina sem raðar í stafrófsröð eftir nafni fyrst og námskeiði síðan.

Vantar meira pláss? ☐ Svarið er að finna _____

- (b) (6 stig) Skrifðu kóða til að finna hve margir nemendur eru skráðir í a.m.k. fjögur námskeið. Hér má ekki nota neinar aðrar gagnagrindur, eingöngu fylkið a . Tíminn á aðferðinni má ekki vera $O(N^2)$ þar sem N er fjöldi staka.

Vantar meira pláss? ☐ Svarið er að finna _____

- (c) (5 stig) Skrifðu kóða sem finnur þau námskeið sem eru með 30 nemendur eða fleiri skráða (hér má nota hvaða gagnagrind sem er).

Vantar meira pláss? ☐ Svarið er að finna _____

10. (7 stig) Gefið er stefnt óhringað net (e. Directed Acyclic Graph, DAG) G með viktir w á leggjunum. Lýsið reikniriti sem finnur þá leið á milli s og t sem hefur stystu meðaltalslengd. (Vísbending: geymið upplýsingar í hverjum hnúti um stystu leið eftir fjölda leggja).

Vantar meira pláss? ☐ Svarið er að finna _____

(Auð síða ef aukapláss vantar)

```
public class Stack<Item> implements Iterable<Item>
```

Stack()	<i>create an empty stack</i>
void push(Item item)	<i>add an item</i>
Item pop()	<i>remove the most recently added item</i>
boolean isEmpty()	<i>is the stack empty?</i>
int size()	<i>number of items in the stack</i>

```
public class MaxPQ<Key> extends Comparable<Key>>
```

MaxPQ()	<i>create a priority queue</i>
MaxPQ(int max)	<i>create a priority queue of initial capacity max</i>
MaxPQ(Key[] a)	<i>create a priority queue from the keys in a[]</i>
void insert(Key v)	<i>insert a key into the priority queue</i>
Key max()	<i>return the largest key</i>
Key delMax()	<i>return and remove the largest key</i>
boolean isEmpty()	<i>is the priority queue empty?</i>
int size()	<i>number of keys in the priority queue</i>

```
public class Graph
```

Graph(int V)	<i>create a V-vertex graph with no edges</i>
Graph(In in)	<i>read a graph from input stream in</i>
int V()	<i>number of vertices</i>
int E()	<i>number of edges</i>
void addEdge(int v, int w)	<i>add edge v-w to this graph</i>
Iterable<Integer> adj(int v)	<i>vertices adjacent to v</i>
String toString()	<i>string representation</i>

```
public class Digraph
```

Digraph(int V)	<i>create a V-vertex digraph with no edges</i>
Digraph(In in)	<i>read a digraph from input stream in</i>
int V()	<i>number of vertices</i>
int E()	<i>number of edges</i>
void addEdge(int v, int w)	<i>add edge v->w to this digraph</i>
Iterable<Integer> adj(int v)	<i>vertices connected to v by edges pointing from v</i>
Digraph reverse()	<i>reverse of this digraph</i>
String toString()	<i>string representation</i>

```
public class Topological
```

Topological(Digraph G)	<i>topological-sorting constructor</i>
boolean isDAG()	<i>is G a DAG?</i>
Iterable<Integer> order()	<i>vertices in topological order</i>

```
public class EdgeWeightedDigraph
```

EdgeWeightedDigraph(int V)	<i>empty V-vertex digraph</i>
EdgeWeightedDigraph(In in)	<i>construct from in</i>
int V()	<i>number of vertices</i>
int E()	<i>number of edges</i>
void addEdge(DirectedEdge e)	<i>add e to this digraph</i>
Iterable<DirectedEdge> adj(int v)	<i>edges pointing from v</i>
Iterable<DirectedEdge> edges()	<i>all edges in this digraph</i>
String toString()	<i>string representation</i>

```
public class UF
```

UF(int N)	<i>initialize N sites with integer names (0 to N-1)</i>
void union(int p, int q)	<i>add connection between p and q</i>
int find(int p)	<i>component identifier for p (0 to N-1)</i>
boolean connected(int p, int q)	<i>return true if p and q are in the same component</i>
int count()	<i>number of components</i>

```
public class ST<Key, Value>
```

ST()	<i>create a symbol table</i>
void put(Key key, Value val)	<i>put key-value pair into the table (remove key from table if value is null)</i>
Value get(Key key)	<i>value paired with key (null if key is absent)</i>
void delete(Key key)	<i>remove key (and its value) from table</i>
boolean contains(Key key)	<i>is there a value paired with key?</i>
boolean isEmpty()	<i>is the table empty?</i>
int size()	<i>number of key-value pairs in the table</i>
Iterable<Key> keys()	<i>all the keys in the table</i>

```
public class Paths
```

Paths(Graph G, int s)	<i>find paths in G from source s</i>
boolean hasPathTo(int v)	<i>is there a path from s to v?</i>
Iterable<Integer> pathTo(int v)	<i>path from s to v; null if no such path</i>

```
public class DirectedDFS
```

DirectedDFS(Digraph G, int s)	<i>find vertices in G that are reachable from s</i>
DirectedDFS(Digraph G, Iterable<Integer> sources)	<i>find vertices in G that are reachable from sources</i>
boolean marked(int v)	<i>is v reachable?</i>

```
public class DirectedCycle
```

DirectedCycle(Digraph G)	<i>cycle-finding constructor</i>
boolean hasCycle()	<i>does G have a directed cycle?</i>
Iterable<Integer> cycle()	<i>vertices on a cycle (if one exists)</i>

```
public class DirectedEdge
```

DirectedEdge(int v, int w, double weight)	
double weight()	<i>weight of this edge</i>
int from()	<i>vertex this edge points from</i>
int to()	<i>vertex this edge points to</i>
String toString()	<i>string representation</i>

```
public class SP
```

SP(EdgeWeightedDigraph G, int s)	<i>constructor</i>
double distTo(int v)	<i>distance from s to v, ∞ if no path</i>
boolean hasPathTo(int v)	<i>path from s to v?</i>
Iterable<DirectedEdge> pathTo(int v)	<i>path from s to v, null if none</i>