1. (3 pont) Írja le saját szavaival, hogy hogyan definiálunk egy absztrakt adattípust. Vagyis mit kell specifikáljunk ahhoz, hogy az egy absztrakt adattípus leírás legyen?

*Megkell adni, hogy milyen értékek vannak benne, és meg kell adni a műveleteket, amiket el lehet végezni, hogy mikor és mit lehet elvégezni és ez mit fog jelenteni, ha elvégezzük a műveletet.*

2. (3 pont) Miért használunk absztrakt adattípusokat? Mik az absztrakt adattípusok céljai?

*Azokkal a részletekkel foglalkozunk, ami egy felhasználónak fontos. A fő hozzáadott értéke, hogy teljesen elválik az implementációjától. A fő cél a felcserélhetőségi szempont. Tehát ha definiálunk egy absztrakt adattípus és erre építünk egy algoritmust, akkor utána minden olyan adat ami ennek az adattípusnak megfelel és tudja a megfelelő műveleteket, akkor arra az algoritmusunk működni fog.*

*Előnyei: egységbezárás (műveleteket csoportosítja, implementációfüggetlen gondolkodásmód), flexibilitás (jobb implementációra való csere lehetősége).*

3. (6 pont) Nevezzen meg a kurzus során tanult absztrakt adattípusokat.

*Dinamikus tömb, halmaz, listák, asszociatív tömb vagy szótár, verem, sor, fa, gráf.*

4. (3 pont) Gondolja végig, hogy hogyan működik egy láncolt lista segítségével implementált dinamikus tömb és a hagyományos tömb segítségével, tömb duplázással, implementált dinamikus tömb. Egy nagy dinamikus tömb felépítése során az idő egyik részében az egyik adatszerkezet foglal több memóriát, máskor a másik. Írja le, hogy mi lehet az oka váltásnak?

Mikor fordulhat elő, hogy a láncolt listás dinamikus tömb foglal több helyet?

*Ha primitív az absztrakt adattípus, például szám, akkor a láncolt listás implementációhoz 2/3-al több tárhely szükséges az adminisztráció miatt.*

Mikor fordulhat elő, hogy a tömbös dinamikus tömb foglal több helyet?

*Ha a dinamikus tömb nagyon nagy méretű és a tárolására nem elegendő sem a processzor memóriája, sem az L2, akkor egy elem lekérdezésekor tömbösen lesznek mozgatva az adatok a további tárhelyekről, a gyorsabb munkavégzés miatt.*

5. (10 pont) Definiálja a karakterláncokat tartalmazó halmaz adattípusát Java nyelven (írja meg a classt vagy classokat amik ehhez szükségesek, a metódusoknak a törzsét nem kell megadni, azt elég csak {…}-al jelölni, nevük, előtagjaik, visszatérési értékük és paramétereik azonban kellenek).

class halmaz{

private String[] karakterlanc;

public halmaz(){karakterlanc=new String[0]}

public boolean elemHozzadasa(String elem){}

public String elemKivetele(){}

public int szamossag(){}

public halmaz metszet(halmaz masikHalmaz){}

public halmaz egyesites(halmaz masikHalmaz){}

public halmaz kivonas(halmaz masikHalmaz){}

public boolean uresE(){}

public void urites(){}

public boolean tartalmazzaE(String elem){}

}

6. (2 pont) Láncolt listák esetén mi a különbség a külső és belső adattárolás között (gondoljon arra, hogy a listában tárolt adat összetett)?

*A belső adattárolásnál az adott adattípusban vannak eltárolva az adatok, míg a külső esetében valamilyen más adattípusban. rajz*

7. (2 pont) Nevezze meg az asszociatív tömb adattípus két legfontosabb műveletét.

*Létrehozás, hozzáadás.*

8. (3 pont) Egy veremmel a következő műveletsort végezzük el: A\*BC\*\*DAA\*\*F\*H, balról jobbra haladva. A nyomtatott nagybetűs karakterek esetén az adott karaktert push-oljuk a verembe, \* a popnak felel meg. Milyen sorrendben jönnek ki karakterek a veremből, illetve mi lesz a verem végállapota a műveletsor után (olyan sorrendben ahogy elhagynák a vermet)?

Kimeneti sorrend: *ACBAAF* Végállapot: *DH*

9. (1 pont) Nevezzen meg olyan nyelvi adattípusokat, amik láncolt listát valósítanak meg?

*Verem, sor.*

10. (2 pont) Az asszociatív tömbök párokat tartalmaznak. Hogyan hívjuk a valami1-et és valami2-t? valami1: *kulcs* valami2: *érték*

11. (2 pont) Mi az asszociatív tömbök legfontosabb tulajdonsága (gondoljon a műveletek komplexitására a hagyományos tömbökhöz képest)? *O(1) – minden művelete.*

12. (3 pont) Egy gráf szomszédsági mátrixszal adott

[

0 1 0 0 0 1

1 0 0 0 1 0

0 0 1 1 0 1

0 0 0 0 0 0

0 0 0 0 1 0

0 0 1 1 0 0

]

a. (2 pont) Irányított vagy irányítatlan gráfot definiál a fenti szomszédsági mátrix? Válaszát indokolja.

*Irányított, mert ellenkező esetben a szomszédsági mátrixnak szimmetrikusnak kellene legyen.*

b. (1 pont) Rajzolja fel a megfelelő gráfot