GRAFOAK: ARIKETAK

1) **Eman dezagun G= {N, A} grafo zuzendu bat dugula, honako auzokidetasun-listarekin adierazia:**

   N, adabegien (erpinen) multzoa: {1, 2, 3, 4, 5, 6, 7, 8}
   A arkuen multzoa adierazten duen AL auzokidetasun-lista:
   AL[1] = {4}
   AL[2] = {1}
   AL[3] = {8}
   AL[4] = {2,3,8}
   AL[5] = {1,2,6}
   AL[6] = {2,7}
   AL[7] = {3}
   AL[8] = {7}

   a) Idatzi G grafoko adabegiak bisitatutako ordenan, baldin eta sakonerako korritzea egingo bagenu, 1 adabegian hasita.
   b) Idatzi G grafoko adabegiak bisitatutako ordenan, baldin eta zabalerako korritzea egingo bagenu, 1 adabegian hasita.

2) **Idatzi eta analizatu algoritmo bat, sakonerako ibilbidea erabiliz, indegree(1..n) eta outdegree(1..n) bektoreak kalkulatuko dituena. Grafoko adabegi bakoitzeko:**
   indegree(i)= i adabegira iristen den arkuen kopurua
   outdegree(i)= i adabegitik ateratzen diren arkuen kopurua

3) **Eman dezagun ordenagailu-sare bat dugula. X ordenagailu bakoitza bere auzoko Y ordenagailu bakoitzarekin komunika daiteke euro 1eko prezioan. Idatzi algoritmo bat, zeinak prezioa(1..n) taula kalkulatuko duen:**
   prezioa(i) = 1 ordenagailuak i ordenagailuarekin konektatzeko ordaindu beharreko prezio minimoa izanik.

4) **Idatzi eta analizatu algoritmo bat, G=(N,A) grafo zuzendu bat emanik, auzokidetasun-listekin definitua, grafo hori zuhaitz bat den ala ez erabakitzen duena. A adabegitik B adabegira arku bat badago, A B-ren aita dela suposatu beharko dugu.**

5) **Emanik G=(N,A) grafo ez-zuzendu bat, eta pisu bat daukana adabegi bakoitzean (kontuz!, ez arkuan, adabegian baizik!), idatzi algoritmo bat zeinak grafoko biderik luzeena kalkulatzen duen eta bide hori hasten den adabegia itzultzen duen. Kontuan izan adabegi batetik bestera joateko beti pisu handiena daukana aukeratuko dela.**