

1. **Winged edge štruktúra** – implementujte túto štruktúru, zadávanie vrcholov a hrán ako dvojicu indexov na vrcholy a následné naplnenie ostatných položiek v štruktúre. Zobrazte zadanú štruktúru ako 2D graf. Pre vybranú stenu vyhladajte všetky susedné steny. (*Machovic*)
2. **Quad edge štruktúra** – implementujte túto štruktúru, zadávanie vrcholov a hrán ako dvojicu indexov na vrcholy a následné naplnenie ostatných položiek v štruktúre. Zobrazte zadanú štruktúru ako 2D graf. Pre vybraný vrchol nájdite susedné hrany. (*Gieci*)
3. **Half edge štruktúra** – implementujte túto štruktúru, zadávanie vrcholov a hrán ako dvojicu indexov na vrcholy a následné naplnenie ostatných položiek v štruktúre. Zobrazte zadanú štruktúru ako 2D graf. Pre vybranú hranu nájdite všetky susedné hrany. (*Tencer*)
4. **Binary space partition 2D** – naimplementujte zadávanie úsečiek ako dvojicu dvojrozmerných bodov a nakoniec auto-prerozdelenie priestoru do binárneho stromu. Usecky a rozdelenie priestoru zobrazte. Pre danú pozíciu vypíšte front-to-back usecky. (*Reznakova*)
5. **Strom AAB obalok** – naimplementujte zadávanie trojuholníkov ako trojice dvojrozmerných bodov a vytvorenie stromu obalok pre dané trojuholníky. Trojuholníky a aj jednotlivé obalky zobrazte. Nakoniec naimplementujte algoritmus na zistenie trojuholníkov, ktoré ležia v danej polrovine. (*Maricak*)
6. **Rovinný graf so susednosťou vrcholov** – naimplementujte zadávanie rovinného grafu pomocou zadania súradníc a pre každý vrchol zoznam indexov susedných vrcholov. Štruktúru zobrazte a pre dva zadané vrcholy nájdite cestu z jedného do druhého vrchola. (*Rejda*)
7. **Strom sferických obalok** - naimplementujte zadávanie úsečiek ako dvojicu dvojrozmerných bodov a vytvorenie stromu obalok pre dané usecky. Usecky aj jednotlivé obalky zobrazte. Potom pre zadanú priamku je potrebné najst všetky priesečky priamky a usečiek pomocou stromu. (*Tirinda*)
8. **Quadtree** – naimplementujte zadávanie bodov ako dvojicu súradníc a vytvorte quadtree tak, aby každý list obsahoval max 1 bod. Potom na tejto štruktúre urobte vyhľadávanie bodov, ktoré patria do zadaného obdĺžnika. (*Kaciak*)
9. **Segment Tree** – naimplementujte zadávanie usečiek ako dvojice bodov a vytvorte segment tree. Potom na tejto štruktúre urobte vyhľadávanie prieniku zadaných zvislej usecky so zadanými usečkami pomocou segment tree. (*Fabo*)
10. **kD Tree** – naimplementujte zadávanie bodov a vytvorte kD tree. Potom na tejto štruktúre urobte vyhľadávanie bodov, ktoré patria danému obdĺžniku. (*Pacal*)
11. **Distance Field** – zadaním uzavretej lomenej čiary sa automaticky vyplní mriežka s oznamienkovanou vzdialenosťou. Toto vzdialenostné pole sa potom farebne zobrazí. Naimplementuje sa nájdenie vzdialenosti daného bodu od lomenej čiary pomocou pola. (*Jesko*)
12. **Implicitná krivka** – pre implicitne zadanú krivku v 2D vytvorte jej aproximáciu v podobe lomenej čiary. (*Hrabovec*)
13. **Interval Tree** – Pre zadané intervaly na usečke vytvorte intervalový strom a pre danú hodnotu nájdite všetky intervaly, ktoré obsahujú danú hodnotu. (*Cajagi*)
14. **Quadtree 2** – pre zadanú množinu trojuholníkov vytvorte quadtree a použite ho na zistenie všetkých trojuholníkov, ktoré obsahujú daný bod. (*Apalovic*)
15. **Point Clouds** – pre zadanú množinu bodov vypočítajte lomenu čiaru, ktorá dobre aproximuje zadané body. Štruktúry vykreslite.
16. **Gabrielov graf a Relatívny graf susednosti** - pre zadanú množinu bodov vytvorte a vykreslite dva druhy geometrických grafov. (*Kolek*)

Kazdy priklad by mal obsahovat interaktivne zadavanie danyh udajov a vykreslenie vystupu.