1. **tétel**

***A C programozási nyelv I.:*** *Adattípusok, deklarációik, feltételes utasítások.****Általános ismeretek:*** *adat és információ, entrópia fajtái, kifejezések infix és postfix alakja.****Mesterséges intelligencia:*** *Keresési problémák állapottér-reprezentációja, példák.  
Neminformált keresési eljárások (mélységi, szélességi, optimális).*

**Elemi adattípusok:** int, char, float, double  
**Összetett adattípusok:** tömb, struktúra  
**pointer típus:** a memóriacímet tartalmazza

Deklaráláskor a rendszer lefoglalja a típusnak megfelelő memória területet. A const nem változik.

**if-else:** elágazás  
**switch-case:** többfelé történő elágazás

**Adat:** Olyan jelsorozat, amely alkalmas a feldolgozásra  
**Információ:** Értelmezett adat  
**Entrópia:** Egy jelsorozat információtartalmát fejezi ki.

**Minimális entrópia:** mindig ugyanaz történik: 0  
**Maximális entrópia:** ha a bekövetkezési esélyek azonosak: log2n  
 **Relatív entrópia:** A tényleges entrópia és a maximális hányadosa

**infix:** Az operátorokat az operandusok közé tesszük  
**postfix:** Az operátorokat az operandusok után használjuk

**Állapottér reprezentáció:** Egy probléma megoldásához szükséges tér, tulajdonságok és jellemzők modellezése. Tartalmazza az összes lehetséges állapotot. A megoldáshoz a **kezdőállapotból** valamelyik **célállapotba** kell eljutni. Az állapotok változását **operátorokkal (műveletekkel)** érjük el.  
**Költség:** az operátor használatának költsége  
**Előfeltételek:** milyen állapotban használható az adott operátor

**Nem informált keresési eljárások:** Nincs információ az állapotokról.

**Szélességi keresés:** Minden csomópontot megvizsgál, mielőtt a következő szintre lépne. FIFO (sor). Idő- és tárigénye nagy. Teljes és ha minden műveletnek ugyanannyi a költsége, akkor optimális is.

**Mélységi keresés:** A keresési fa mélyebb csomópontjai felé halad. LIFO (verem). Időigénye nagy. Tárigény gráf esetén annyi, mint az időigény. Fakeresésnél lineáris tár. Ha az állapottér végtelen, akkor nem teljes. Nem optimális.

**Optimális kereső:** Minden csúcshoz tárolja az út költségét. Mindig a kisebb költségű lépést választja. Teljes és optimális.

1. **tétel**

***A C programozási nyelv II.****: Ciklusszervezési lehetőségek, függvénykezelés, paraméterkiértékelés, hatáskörkezelés (statikus, dinamikus).****Általános ismeretek:*** *számrendszerek, számábrázolás (fix és lebegőpontos), karakter, szöveg és logikai adat ábrázolása.****Mesterséges intelligencia:*** *A heurisztika fogalma, példák. A\* algoritmus. Az A\* algoritmus teljessége. Kétszemélyes, teljes információjú, determinisztikus játékok: a stratégia fogalma, minimax-algoritmus, alfa-béta vágás.*

Elő és utótesztelő ciklusok.  
**for:** inicializáció és feltétel, majd a törzs végrehajtása után léptetés  
**while:** nincs init csak feltétel, a léptetés a törzsben van  
**do-while:** egyszer mindenképpen végre lesz hajtva a törzs

**függvények:** többször meghívható alprogramok. A visszatérési értéküket a legelején kell megadni. A paraméterek lokálisak lesznek. A paraméterek átadása érték szerint történik.

**lokális változó:** csak a blokkon belül érhető el  
**globális változó:** a függvényeken kívül vannak deklarálva, ezért mindenhol elérhetőek

**statikus:** csak egyszer jönnek létre és egyszer kapnak értéket

**fix pontos ábrázolás:** egész számokhoz (kettes komplemens)  
**lebegőpontos ábrázolás:** törtekhez (előjel, mantissza, kettedes eltolás)  
**karakter ábrázolás**: kódtáblák (ASCII, Unicode)  
**szöveg ábrázolása:** karakter tömbben, kell egy \0 zárókarakter is  
**logikai érték ábrázolása:** 1byte-on, 1 vagy 0

**Heurisztika:** Egy állapot költségének becslése. Nem becsüli túl a költséget. A célállapot értéke 0.

**A\* algoritmus:** A legolcsóbb utat keresi. A megtett út költsége és a célba jutás költségének összege. Teljes és optimális.

A **kétszemélyes játékok** kompetitív multiágens környezetek.  
**Teljes információjú játék:** Minden eleméről információnk van.  
**Determinisztikus:** Nincs szerepe a véletlennek.

**Stratégia:** Minden játékálláshoz meghatározza a lépéseket.  
**Nyerő stratégia:** Az ellenfél lépéseitől függetlenül győzelemre vezet  
**Nem-vesztő stratégia:** Az ellenfél lépéseitől függetlenül, nem vezet vereséghez.

**minimax:** A döntést a minimum és maximum keresés eredményéből hozza meg.  
**alfa-béta:** Az eredménye ugyanaz, mint a minimax-nak, de a szükségtelen ágakat nem járja be. Az alfa maximumot, a béta minimumot keres.