## ZH megoldás

### B/1 feladat: Fejelem nélküli, egyirányú lista maximumkiválasztásos rendezése

Két lista van: rendezett, rendezetlen. Rendezetlenből max elemet átfűzzük a rendezettbe. (Üres listára is működni fog.)

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
|  | | |  |
|  | | | |
|  |  | | |
|  | | |
|  |  | |
|  | | |

Maxkivétel:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
|  | | | |  |
|  | | | | |
|  | | | | |
|  |  | | | |
|  |  | | |
|  | | | |
|  | | | | |
|  | | |  | |

### B/2 feladat: Monoton növően rendezett vektorból szigorúan monoton rendezett sorozatot készíteni és visszaadni egy hosszt.

Gyakori hiba: Sokan balra shiftelték a tömböt az ismétlődéseknél, de ez nagyon költséges. A tömb végét () nem kellene módosítani.

### A/1 feladat: tömbből fejelemes egyirányú ciklikus listát kell építeni

Nem volt nehéz, de a lista végére mutatót kell nyilvántartani, amit sokan elfelejtettek.

### A/2 feladat: Szimmetrikus listát monoton növő szakaszokra kellett bontani.

Ami miatt -1 pontot kaptam: helyett kell, mert két irányban láncolt lista volt a feladatban.

## Listából halmaz (Ismétlődő elemek kihagyása)

Mesterséges intelligencia feladat során felmerülhet, hogy sok a bemenet és minden megoldásra szükség van, de többször is kijöhet ugyan az a megoldás.

A megoldások rendezett beillesztéssel való tárolása négyzetes futásidejű lenne, nem jó.

Ha összegyűjtjük az elemeket és utána rendezzük, majdnem ugyan olyan rossz. Tehát mindenféleképpen négyzetes lenne a futásidő, ami iszonyatosan gyorsan nő.

Mi a megoldás:

* Vesszük a sorozatot, elfelezzük. (Lineáris)
* Külön-külön elvégezzük az elem-duplikáció-megszüntetést. (Részfeladat, nem mi foglalkozunk vele.)
* A két sorozatot rendezetten összeuniózzuk. (Lineáris)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | |  | | | |  |
|  | |  | |  | |  | |  |
|  |  |  |  |  |  |  |  |  |

A két félre rekurzívan meghívjuk ugyan ezt a feladatot.

Tehát:

Ez mennyivel jobb, mint ? Sokkal.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

## Uniórendezés (Union Sort)

|  |  |
| --- | --- |
|  | |
|  |  |
|  | |

:

|  |  |  |
| --- | --- | --- |
|  | | |
|  | |  |
|  | | |
|  |  | |

Feltesszük, hogy egyik lista-fél sem üres.

### A Merge Sort hasonló

|  |  |
| --- | --- |
|  | |
|  |  |
|  | |

:

|  |  |  |
| --- | --- | --- |
|  | | |
|  | |  |
|  | | |
|  |  | |

Mindkettőre:

### Divide

|  |  |  |
| --- | --- | --- |
|  | | |
|  | |  |
|  | | |
|  | | |
|  |  | |
|  | | |

### Önmagát hívó eljárás (rekurzió) nem gond

|  |  |  |  |
| --- | --- | --- | --- |
| CODE  Program kódját ide töltjük be | GLOB  Globális változók | CALL STACK  Eljáráshívásokat adminisztrálja | DYNAMIC HEAP  A utasítások innen veszik a memóriát |

Call stack:

|  |
| --- |
|  |
|  |
|  |

Call frame:

|  |
| --- |
| Formal parameters |
| Local variables |
| Return address |

Gyakorlat vége.