**Ex 1:** Given six memory partitions of 300 KB, 600 KB, 350 KB, 200 KB, 750 KB, and 125 KB (in order), how would the first-fit, best-fit, and worst-fit algorithms place processes of size 115 KB, 500 KB, 358 KB, 200 KB, and 375 KB (in order)? Rank the algorithms in terms of how efficiently they use memory.

* **First fit:** Allocate the first hole that is big enough. Searching can start either at the beginning of the set of holes or at the location where the previous first-fit search ended. We can stop searching as soon as we find a free hole that is large enough.
* **Best fit:** Allocate the smallest hole that is big enough. We must search the entire list, unless the list is ordered by size. This strategy produces the smallest leftover hole.
* **Worst fit:** Allocate the largest hole. Again, we must search the entire list, unless it is sorted by size. This strategy produces the largest leftover hole, which may be more useful than the smaller leftover hole from a best-fit approach.
  + M1=300 KB M2=600 KB M3=350 KB M4=200 KB M5=750 KB M6=125 KB
  + P1=115 KB P2=500 KB P3=358 KB P4=200 KB P5=375 KB

**First-fit**

– P1 = 115 M1=**300** M2=600 M3=350 M4=200 M5=750 M6=125

– P2 = 500 M1=185 M2=**600** M3=350 M4=200 M5=750 M6=125

– P3 = 358 M1=185 M2=100 M3=350 M4=200 M5=**750** M6=125

– P4 = 200 M1=185 M2=100 M3=**350** M4=200 M5=392 M6=125

– P5 = 375 M1=185 M2=100 M3=150 M4=200 M5=**392** M6=125

**Best-fit**

– P1 = 115 M1=300 M2=600 M3=350 M4=200 M5=750 M6=**125**

– P2 = 500 M1=300 M2=**600** M3=350 M4=200 M5=750 M6=10

– P3 = 358 M1=300 M2=100 M3=350 M4=200 M5=**750** M6=10

– P4 = 200 M1=300 M2=100 M3=350 M4=**200** M5=392 M6=10

– P5 = 375 M1=300 M2=100 M3=350 M4=000 M5=**392** M6=10

**Worst-fit**

– P1 = 115 M1=300 M2=600M3=350 M4=200 M5=**750** M6=125

– P2 = 500 M1=300 M2=600 M3=350 M4=200 M5=**635** M6=125

– P3 = 358 M1=300 M2=**600** M3=350 M4=200 M5=135 M6=125

– P4 = 200 M1=300 M2=242 M3=**350** M4=200 M5=135 M6=125

– P5 = 375 M1=300 M2=242 M3=150 M4=200 M5=135 M6=125

*Rank the algorithms in terms of how efficiently they use memory:*

Best-fit > First-fit > Worst-fit

**Ex 2:** Student write a short report that compares the advantages as well as disadvantages of the allocation algorithms, namely First-Fit, Best-Fit, Worst-Fit.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **First-fit** | **Best-fit** | **Worst-fit** |
| Advantages | It’s very fast in allocation because searching or sorting is not required.  Algorithm is very simple to implement (Just find a first holes which is enough space) | Space wastage is very less or negligible.  Every Time it finds the appropriate hole to allocate file. | It’s also fast but slower than first fit.  It’s algorithm is simple,so it easy to implement. |
| Disadvantages | Space maybe wasted because no searching for fit space. | It’s slower than First-fit because of searching (or sorting or both of th)  Algorithm is difficult to implement. | Large amount of space may be wasted.  It requires sorting and searching free holes.  It consumes time.  It does not provides the optimal solution. |