

## First Fit

In the first fit approach is to allocate the first free partition or hole large enough which can accommodate the process. It finishes after finding the first suitable free partition.

## Advantage

Fastest algorithm because it searches as little as possible.

## Disadvantage

The remaining unused memory areas left after allocation become waste if it is too smaller. Thus request for larger memory requirement cannot be accomplished.

```
first fit simulation = 1
next fit simulation = 2
best fit simulation = 3
worst fit simulation = 4
enter your choice
1
9999 9999 9999 9999 9999 9999 0 0 0 9982 9982 9982 9982 9982 9982 9982 9982 9983 9983 9983 9983 9983 9983 9983 9986
9986 9986 9986 9986 9986 9986 9986 9986 9987 9987 9987 9988 9988 9988 9988 9988 9988 9990 9990 9990 9990 9990 9990 9998 9
998 9998 9998 9998 9998 9998 9998 9998 9998 0 0 9984 9984 9984 9984 9992 9992 9997 9997 9997 9997 9997 9997 0 0 0 0
9989 9989 9989 9989 9989 9989 9989 9989 9991 9991 9991 9991 9993 9993 9993 9993 9993 9995 9995 9995 9996 9996 9
996 10000 10000 10000 10000 10000 10000 10000 10000 0 0 0 0 9985 9985 9985 9985 9985 9985
There are 15 fragments in memory.
Allocation time: 0.0123214 s
```

## Best Fit

The best fit deals with allocating the smallest free partition which meets the requirement of the requesting process. This algorithm first searches the entire list of free partitions and considers the smallest hole that is adequate. It then tries to find a hole which is close to actual process size needed.

## Advantage

Memory utilization is much better than first fit as it searches the smallest free partition first available.

## Disadvantage

It is slower and may even tend to fill up memory with tiny useless holes.

```

Process 2 is stored in999921
Process 2 is stored in999922
Process 2 is stored in999923
Process 2 is stored in1000013
Process 2 is stored in1000014
Process 2 is stored in1000015
Process 2 is stored in1000016
Process 2 is stored in1000017
Process 2 is stored in1000018
Process 2 is stored in1000019
Process 2 is stored in1000020
Process 2 is stored in1000021
Process 2 is stored in1000022
9998 9998 9998 9998 9998 9998 9998 9997 9997 9997 9997 9997 10000 10000 10000 10000 10000 10000 10000 10000 10000 9999 9996 9996 9996 9
996 9996 9996 9971 9972 9972 9972 9972 9972 9972 9972 9972 9972 9975 9975 9975 9975 9975 9976 9976 9976 9976 9976 9976 9995 9995 9995 99
91 9991 9991 9991 9991 9991 9992 9992 9992 9992 9992 9992 9992 9992 9992 9992 9994 9994 9994 9993 9993 9993 9993 9988 9988 9988 9988 998
7 9987 9987 9987 9987 9987 9987 9980 9980 9980 9979 9979 9979 9979 9977 9977 9977 9977 9985 9985 9985 9985 9985 9985 9985 9985 9985 9984
9984 9984 9984 9983 9983 9983 9981
There are 0 fragments in memory.
Allocation time: 2.94846 s
*** stack smashing detected ***: /home/a.out terminated
Aborted

```

```

9998 9998 9998 9998 9998 9998 9998 9998 9997 9997 9997 9997 9997 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 9999 9996 9996 9996 9
996 9996 9996 9971 9972 9972 9972 9972 9972 9972 9972 9972 9972 9975 9975 9975 9975 9975 9976 9976 9976 9976 9976 9976 9995 9995 9995 99
91 9991 9991 9991 9991 9991 9992 9992 9992 9992 9992 9992 9992 9992 9992 9992 9994 9994 9994 9993 9993 9993 9993 9988 9988 9988 9988 998
7 9987 9987 9987 9987 9987 9987 9980 9980 9980 9979 9979 9979 9979 9977 9977 9977 9977 9985 9985 9985 9985 9985 9985 9985 9985 9985 9984
9984 9984 9984 9983 9983 9983 9981
There are 0 fragments in memory.
Allocation time: 2.94846 s
*** stack smashing detected ***: /home/a.out terminated
Aborted

```

## Worst fit

In worst fit approach is to locate largest available free portion so that the portion left will be big enough to be useful. It is the reverse of best fit.

## Advantage

Reduces the rate of production of small gaps.

## Disadvantage

If a process requiring larger memory arrives at a later stage then it cannot be accommodated as the largest hole is already split and occupied.

```

9996 9996 9996 9996 9996 9996 9996 9996 9996 9997 9997 9997 9997 9997 9997 9997 9998 9998 9998 9998 9999 9999 9999 9999 10000 10000
10000 10000 10000 10000 10000 10000 10000 10000 10000 9982 9983 9983 9983 9984 9984 9984 9984 9984 9984 9984 9985 9985 9985 9985 9986 9986 99
86 9986 9986 9987 9987 9987 9987 9988 9988 9988 9988 9988 9988 9988 9988 9989 9989 9989 9989 9989 9989 9989 9990 9990 9990 9990 9990 9990 999
0 9990 9991 9991 9991 9991 9991 9991 9991 9992 9992 9992 9992 9992 9992 9992 9992 9993 9993 9993 9993 9993 9993 9993 9994 9994 9994 9994
9994 9994 9994 9994 9995 9995 9995
There are 0 fragments in memory.
Allocation time: 0.622632 s
...Program finished with exit code 0

```

## Next fit

Next fit is a modified version of first fit. It begins as first fit to find a free partition. When called next time it starts searching from where it left off, not from the beginning.

```

first fit simulation = 1
next fit simulation = 2
best fit simulation = 3
worst fit simulation = 4
enter your choice
2
0 0 0 0 0 0 0 0 0 0 9996 9996 9996 9996 9996 9996 9996 9996 9996 9996 0 9997 9997 9997 0 0 9992 9992 9992 9992 9992 9992 9992 9992 9999 9999 9999 9999
9999 9999 9999 9999 9999 0 10000 10000 10000 10000 10000 10000 10000 10000 10000 0 0 9988 9988 9988 9988 9988 9988 9993 9993 9994 9994 9994 9994 9994 99
94 9994 0 0 0 0 0 0 0 0 9986 9986 9986 9986 9986 9989 9989 9995 9995 9995 9995 9998 9998 9998 9998 9998 9998 9998 9998 9998 0 0 0 9985 998
5 9985 9987 9987 9987 9987 9987 9987 9987 9987 9990 9990 9990 9990 9990 9990 0 9991 9991 9991
There are 33 fragments in memory.
Allocation time: 0.00516258 s

```

## Notes

- Description of the goal was little bit confusing that do I need to fill holes after it get randomly filled and I run my methods or not. I did it this way in the end.
- There wasn't specific point on how to make statistic and request generation. I made them in very simple ways in my main method. But I couldn't print percentage of time.
- My mac out of date, therefore, I used online compiler which might be made it wrong out of my expectation
- It was group project but I...