Azka Yasin

|  |
| --- |
| // jobs sudah terurut berdasarkan profit *descending*  JobSequencing (Job jobs[], n)   1. dmax = 0 2. for i = 0 to n 3. if job[i].deadline > dmax 4. dmax = job[i].deadline 5. for i = 1 to i <= dmax 6. timeslot[i] = -1 7. ~~print dmax~~ 8. for i = 1 to n 9. k = min(dmax, jobs[i-1].deadline) 10. while(k >=1) 11. if timeslot[k] == -1 12. timeslot[k] = i-1 13. filledtimeslot++ 14. break 15. k-- 16. if(filledTimeSlot == dmax) 17. break 18. maxProfit = 0 19. for i = 1 to dmax 20. maxProfit += jobs[timeslot[i]].profit 21. print maxProfit   struct Job {  char id;  int deadline  int profit  }; |

Mutia Rahmi Dewi

|  |  |
| --- | --- |
| // jobs: array of Job, sorted by profit descending  // deadline: array of deadline  // timeslot: array slot schedule  // dmax: max deadline  Job (jobs, dmax, n)   1. for i = 1 to n 2. set k = min(dmax, jobs[i].deadline) 3. while k >= 1 4. if timeslot[k] is EMPTY then 5. timeslot[k] = jobs[i].id 6. break 7. k = k-1 8. totalProfit = 0 9. for i = 1 to dmax 10. totalProfit += jobs[timeslot[i]].profit 11. print totalProfit | struct Job {  int id;  int deadline;  int profit;  }; |