

Job1:  $t_1(8), t_2(3), t_3(5), t_4(2), t_5(4), t_6(3), t_7(7), t_8(5), t_9(6), t_{10}(8), t_{11}(10)$

$t_1 < t_2, t_2 < t_3, t_2 < t_5, t_3 < t_4, t_4 < t_5, t_2 < t_6, t_6 < t_7, t_4 < t_7, t_7 < t_8, t_8 < t_9, t_9 < t_{10}, t_5 < t_{10}$

Job2:  $t_1(3), t_2(2), t_3(8), t_4(4), t_5(5), t_6(1), t_7(6), t_8(3)$

$t_1 < t_2, t_2 < t_3, t_1 < t_4, t_1 < t_3, t_3 < t_5, t_5 < t_6, t_6 < t_7, t_7 < t_8, t_4 < t_7$

Job3:  $t_1(2), t_2(5), t_3(4), t_4(8), t_5(3), t_6(6), t_7(3), t_8(10)$

$t_1 < t_2, t_2 < t_3, t_2 < t_4, t_4 < t_5, t_6 < t_5, t_5 < t_7$

ukoly:

- Sestavte dependency diagram pro Job1 a urcete kritickou cestu. Urcete spodni odhad delky rozvrhu (nejkratsi mozny na nekonecnem mnozství procesoru) a horni odhad delky rozvrhu (nejkratsi mozny na jednom procesoru).
- Sestavte rozvrh pro Job1 na dvou procesorech.
- Sestavte dependency diagram pro Job2 a Job3 a urcete pro ne kritickou cestu. Pro oba pak urcete spodni odhad delky rozvrhu a horni odhad delky rozvrhu.
- Sestavte rozvrh pro J1, J2, J3 na trech procesorech (nahodna, ci libovolna heuristika).
- Spoctete celkovou delku rozvrhu (completion time C) a soucet casu dokonceni jednotlivych Jobu (sum  $C_i$ ).
- Za pouziti vhodne heuristiky sestavte nejkratsi mozny rozvrh (minimalni C) nebo „nejrychlejsi“ (minimalni sum  $C_i$ ).