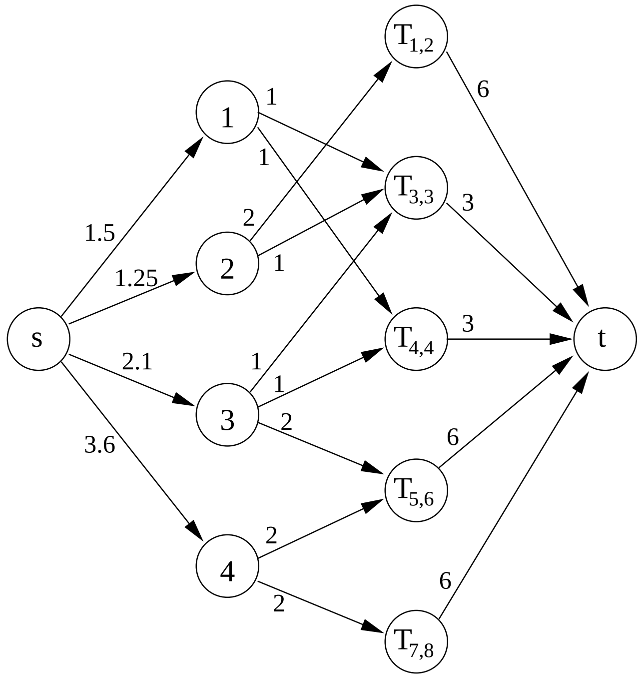
Technische Universität München
Department of Computer Science
Chair for Efficient Algorithms
Prof. Dr. Sami Khuri

WS 1999/2000
Problem Set 4
May 23, 2000

Efficient Algorithms and Data Structures II

Problem 1 (10 credits)

Use the FIFO preflow-push algorithm to produce a maximum flow for the following network.



Recall that this is the network we obtain by casting the problem instance of "Scheduling on uniform parallel machines" that we studied in class into a maximum flow problem. In the diagram, we have 4 jobs with processing times 1.5, 1.25, 2.1, and 3.6, and $T_{k,\ell}$ denotes the interval that starts at the beginning of day k and ends at the beginning of day $\ell+1$.

Problem 2 (10 credits)

If $G = (V, E)$ is a flow network with source s and sink t, f is a preflow in G, and h is a height function, then we say that (u, v) is an *admissible* edge if $c_f(u, v) > 0$, and $h(u) = h(v) + 1$. Otherwise, (u, v) is *inadmissible*. The *admissible network* is $G_{f,h} = (V, E_{f,h})$, where $E_{f,h}$ is the set of admissible edges. Show that $G_{f,h}$ is acyclic.

Problem 3 (20 credits)

The LIFT-TO-FRONT algorithm administers a chained list of nodes from $V - \{s, t\}$ in topological order w.r.t. $G_{f,h}$. Show that the following algorithm computes a maximum flow in time $O(V^3)$.

```
LIFT-TO-FRONT(G, c, s, t)
    INITIALIZE-PREFLOW(G, s)
    L ← V − {s, t} in topological order
    for all v ∈ V − {s, t}
        current[v] ← first node of v's adjacency list
    u ←HEAD(L)
    while u ≠NIL
        old_height ← h(u)
        DISCHARGE(u)
        if h(u) > old_height then
            move u to the front of list L
        u ←NEXT(L, u)
```

Note that INITIALIZE-PREFLOW(G, s) here is similar to the initialization phase of the preflow-push algorithm.