

Day 7 - Instruction graph

Read data

```
In[239]:= SetDirectory[NotebookDirectory[]];
```

```
In[240]:= (in = ReadList["input.txt", String]) ~Take~ 5 // TableForm
```

Out[] // TableForm =

```
Step Z must be finished before step V can begin.
Step V must be finished before step K can begin.
Step M must be finished before step Q can begin.
Step E must be finished before step X can begin.
Step J must be finished before step W can begin.
```

```
In[ ]:= (ex = ReadList["example.txt", String]) // TableForm
```

Out[] // TableForm =

```
Step C must be finished before step A can begin.
Step C must be finished before step F can begin.
Step A must be finished before step B can begin.
Step A must be finished before step D can begin.
Step B must be finished before step E can begin.
Step D must be finished before step E can begin.
Step F must be finished before step E can begin.
```

```
In[ ]:= raw = in; nwork = 5; durationSub = 4;
```

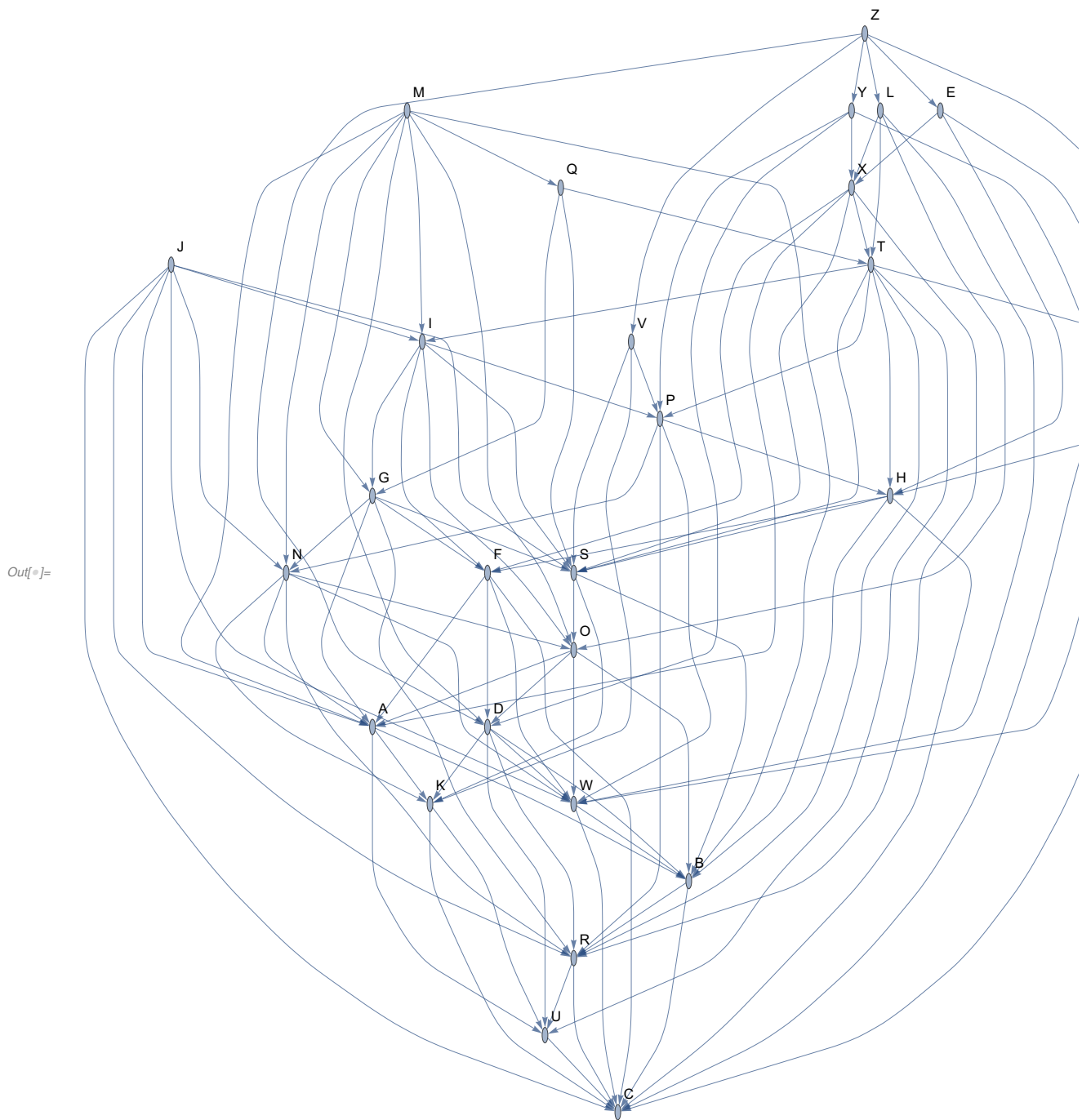
```
In[ ]:= data = (StringSplit /@ raw) [[All, {2, 8}]];
```

```
In[ ]:= edges = DirectedEdge[#[[1]], #[[2]]] & /@ data
```

```
Out[ ]:= {Z → V, V → K, M → Q, E → X, J → W, L → O, Q → T, Y → P, X → R, T → U, I → O, P → H,
G → A, N → A, H → B, F → D, S → O, O → W, D → U, W → B, A → K, B → R, K → C, R → C,
U → C, A → U, J → I, D → K, V → S, H → C, R → U, I → G, D → R, M → B, G → R,
M → I, G → N, M → N, Q → S, I → S, J → R, O → B, G → S, J → C, M → D, T → H,
P → N, S → K, T → C, J → A, G → F, N → R, N → W, T → I, S → B, H → F, B → C,
L → W, N → O, O → A, H → S, F → A, F → C, M → A, Z → H, Z → L, E → H, X → T,
Y → X, E → W, P → R, Z → E, W → C, I → P, X → A, Y → C, I → F, L → T, A → B,
F → W, T → R, X → F, M → O, N → K, T → S, J → N, J → S, O → D, T → P, Z → D,
L → X, Q → G, M → G, P → W, V → P, D → B, Y → D, X → S, K → U, Z → Y, D → W}
```

```
In[ ]:= vertices[edges_] := List@@@edges // Flatten // Union
```

```
In[ ]:= g = Graph[edges, VertexLabels -> "Name", AspectRatio -> 1]
```



Common functions

Find first/last element in graph, idx=1 -> end, idx=2 -> start

```

In[®]:= findOuter[edges_, idx_] := Module[{connected},
  connected = edges[[All, idx]] // Union;
  Complement[vertices[edges], connected] // Sort
]

In[®]:= findOuter[edges, #] & /@ {2, 1}
Out[®]= {{J, M, Z}, {C}}

In[®]:= candidates[edges_, unavailable_, bases_] := Module[{inner, outer},
  inner =
    Select[edges, MemberQ[unavailable, #[[1]]] && ! MemberQ[unavailable, #[[2]]] &];
  outer = Select[bases, ! MemberQ[unavailable, #[[2]]] &];
  Join[inner, outer]
]

In[®]:= checkCandidate[c_, edges_, bases_, unavailable_] := Module[{edgeReq, passing},
  If[c[[1]] === Null,
    ! MemberQ[unavailable, c[[2]]],
    edgeReq = Select[edges, #[[2]] == c[[2]] &];
    passing = Select[edgeReq, MemberQ[unavailable, #[[1]]] &];
    Length[edgeReq] == Length[passing]
  ]
]

```

Part 1

```

In[®]:= walk[edges_, limit_: 5] :=
  Module[{order = {}, bases, clist, done = False, k = 0, consider, light, end},
    bases = DirectedEdge[Null, #] & /@ findOuter[edges, 2];
    end = findOuter[edges, 1];
    While[! done,
      clist = candidates[edges, order, bases];
      consider = Pick[clist,
        checkCandidate[#, edges, bases, order] & /@ clist] ~ SortBy ~ (#[[2]] &);
      light = consider[[1, 2]];
      AppendTo[order, consider[[1, 2]]];
      k++;
      done = light == end[[1]] || k > limit;
    ];
    StringJoin[order]
  ]

In[®]:= walk[edges, 50]
Out[®]= JMQZELVYXTIGPHFNSOADKWBRUC

```

Part 2

```
In[ ]:= bases = DirectedEdge[Null, #] & /@ findOuter[edges, 2]
```

```
Out[ ]:= {Null ↔ J, Null ↔ M, Null ↔ Z}
```

```
In[ ]:= end = findOuter[edges, 1]
```

```
Out[ ]:= {C}
```

```
In[ ]:= duration[char_] := ToCharacterCode[char][[1]] - durationSub
```

```
In[ ]:= duration[#] & /@ {"A", "E", "F"}
```

```
Out[ ]:= {61, 65, 66}
```

General plan : find candidates that can be solved, assign to all free agents and note completion time.
When no more tasks can be assigned (no more tasks or out of workers), mark the first job complete and advance time.

```

In[ ]:= schedule[edges_, nwork_, limit_: 100] := Module[
  {order = {}, bases, clist, done = False, k = 0, consider, light, end, inprogress = {},
   workers, free, node, now = 0, widx, first, sortedWorkers, nodeConsider},
  bases = DirectedEdge[Null, #] & /@ findOuter[edges, 2];
  end = findOuter[edges, 1];
  workers = Association@Table[k → {k, False}, {k, 1, nwork}];
  While[! done,
    clist = candidates[edges, inprogress, bases];
    consider = Pick[clist,
      checkCandidate[#, edges, bases, order] & /@ clist] ~ SortBy ~ (#[[2]] &);
    nodeConsider = Union[#[[2]] & /@ consider];
    free = Select[workers, #[[2]] == False &];
    While[Length[nodeConsider] > 0 && Length[free] > 0 && k++ < limit,
      Print["consider=", nodeConsider];
      node = nodeConsider[[1]];
      nodeConsider = Drop[nodeConsider, 1];
      widx = free[[1, 1]];
      free = Drop[free, 1];
      workers[widx] = {widx, True, node, now + duration[node]};
      AppendTo[inprogress, node];
      Print["workers=", workers];
    ];
    (* Resolve the first worker to be done *)
    sortedWorkers = SortBy[Select[workers, #[[2]] &], #[[4]] &];
    first = sortedWorkers[[1]];
    Print["sorted busy workers=", sortedWorkers];
    now = first[[4]];
    AppendTo[order, first[[3]]];
    widx = first[[1]];
    workers[widx] = {widx, False};
    Print["time=", now, ", added=",
      first[[3]], ", workers=", workers, ", order=", order];
    done = first[[3]] == end[[1]] || k > limit;
  ];
  {StringJoin[order], now}
]

```

```

In[ ]:= schedule[edges, nwork, 1000]

consider={J, M, Z}

workers=<| 1 → {1, True, J, 70}, 2 → {2, False}, 3 → {3, False}, 4 → {4, False}, 5 → {5, False} |>

consider={M, Z}

workers=
  <| 1 → {1, True, J, 70}, 2 → {2, True, M, 73}, 3 → {3, False}, 4 → {4, False}, 5 → {5, False} |>

consider={Z}

```

```

workers=<| 1 → {1, True, J, 70}, 2 → {2, True, M, 73},
  3 → {3, True, Z, 86}, 4 → {4, False}, 5 → {5, False} |>
sorted busy workers=<| 1 → {1, True, J, 70}, 2 → {2, True, M, 73}, 3 → {3, True, Z, 86} |>
time=70, added=J, workers=<| 1 → {1, False}, 2 → {2, True, M, 73},
  3 → {3, True, Z, 86}, 4 → {4, False}, 5 → {5, False} |>, order={J}
sorted busy workers=<| 2 → {2, True, M, 73}, 3 → {3, True, Z, 86} |>
time=73, added=M, workers=<| 1 → {1, False}, 2 → {2, False},
  3 → {3, True, Z, 86}, 4 → {4, False}, 5 → {5, False} |>, order={J, M}
consider={Q}
workers=
  <| 1 → {1, True, Q, 150}, 2 → {2, False}, 3 → {3, True, Z, 86}, 4 → {4, False}, 5 → {5, False} |>
sorted busy workers=<| 3 → {3, True, Z, 86}, 1 → {1, True, Q, 150} |>
time=86, added=Z, workers=<| 1 → {1, True, Q, 150}, 2 → {2, False},
  3 → {3, False}, 4 → {4, False}, 5 → {5, False} |>, order={J, M, Z}
consider={E, L, V, Y}
workers=
  <| 1 → {1, True, Q, 150}, 2 → {2, True, E, 151}, 3 → {3, False}, 4 → {4, False}, 5 → {5, False} |>
consider={L, V, Y}
workers=<| 1 → {1, True, Q, 150}, 2 → {2, True, E, 151},
  3 → {3, True, L, 158}, 4 → {4, False}, 5 → {5, False} |>
consider={V, Y}
workers=<| 1 → {1, True, Q, 150}, 2 → {2, True, E, 151},
  3 → {3, True, L, 158}, 4 → {4, True, V, 168}, 5 → {5, False} |>
consider={Y}
workers=<| 1 → {1, True, Q, 150}, 2 → {2, True, E, 151},
  3 → {3, True, L, 158}, 4 → {4, True, V, 168}, 5 → {5, True, Y, 171} |>
sorted busy workers=<| 1 → {1, True, Q, 150}, 2 → {2, True, E, 151},
  3 → {3, True, L, 158}, 4 → {4, True, V, 168}, 5 → {5, True, Y, 171} |>
time=150, added=Q, workers=<| 1 → {1, False}, 2 → {2, True, E, 151},
  3 → {3, True, L, 158}, 4 → {4, True, V, 168}, 5 → {5, True, Y, 171} |>, order={J, M, Z, Q}
sorted busy workers=
  <| 2 → {2, True, E, 151}, 3 → {3, True, L, 158}, 4 → {4, True, V, 168}, 5 → {5, True, Y, 171} |>
time=151, added=E, workers=<| 1 → {1, False}, 2 → {2, False}, 3 → {3, True, L, 158},
  4 → {4, True, V, 168}, 5 → {5, True, Y, 171} |>, order={J, M, Z, Q, E}
sorted busy workers=<| 3 → {3, True, L, 158}, 4 → {4, True, V, 168}, 5 → {5, True, Y, 171} |>
time=158, added=L, workers=<| 1 → {1, False}, 2 → {2, False}, 3 → {3, False},
  4 → {4, True, V, 168}, 5 → {5, True, Y, 171} |>, order={J, M, Z, Q, E, L}
sorted busy workers=<| 4 → {4, True, V, 168}, 5 → {5, True, Y, 171} |>

```

```

time=168, added=V, workers=
  <|1 → {1, False}, 2 → {2, False}, 3 → {3, False}, 4 → {4, False}, 5 → {5, True, Y, 171}|>
  , order={J, M, Z, Q, E, L, V}
sorted busy workers=<|5 → {5, True, Y, 171}|>
time=171, added=Y, workers=
  <|1 → {1, False}, 2 → {2, False}, 3 → {3, False}, 4 → {4, False}, 5 → {5, False}|>
  , order={J, M, Z, Q, E, L, V, Y}
consider={X}
workers=
  <|1 → {1, True, X, 255}, 2 → {2, False}, 3 → {3, False}, 4 → {4, False}, 5 → {5, False}|>
sorted busy workers=<|1 → {1, True, X, 255}|>
time=255, added=X, workers=
  <|1 → {1, False}, 2 → {2, False}, 3 → {3, False}, 4 → {4, False}, 5 → {5, False}|>
  , order={J, M, Z, Q, E, L, V, Y, X}
consider={T}
workers=
  <|1 → {1, True, T, 335}, 2 → {2, False}, 3 → {3, False}, 4 → {4, False}, 5 → {5, False}|>
sorted busy workers=<|1 → {1, True, T, 335}|>
time=335, added=T, workers=
  <|1 → {1, False}, 2 → {2, False}, 3 → {3, False}, 4 → {4, False}, 5 → {5, False}|>
  , order={J, M, Z, Q, E, L, V, Y, X, T}
consider={I}
workers=
  <|1 → {1, True, I, 404}, 2 → {2, False}, 3 → {3, False}, 4 → {4, False}, 5 → {5, False}|>
sorted busy workers=<|1 → {1, True, I, 404}|>
time=404, added=I, workers=
  <|1 → {1, False}, 2 → {2, False}, 3 → {3, False}, 4 → {4, False}, 5 → {5, False}|>
  , order={J, M, Z, Q, E, L, V, Y, X, T, I}
consider={G, P}
workers=
  <|1 → {1, True, G, 471}, 2 → {2, False}, 3 → {3, False}, 4 → {4, False}, 5 → {5, False}|>
consider={P}
workers=
  <|1 → {1, True, G, 471}, 2 → {2, True, P, 480}, 3 → {3, False}, 4 → {4, False}, 5 → {5, False}|>
sorted busy workers=<|1 → {1, True, G, 471}, 2 → {2, True, P, 480}|>
time=471, added=G, workers=
  <|1 → {1, False}, 2 → {2, True, P, 480}, 3 → {3, False}, 4 → {4, False}, 5 → {5, False}|>
  , order={J, M, Z, Q, E, L, V, Y, X, T, I, G}
sorted busy workers=<|2 → {2, True, P, 480}|>

```

```

time=480, added=P, workers=
  <|1 → {1, False}, 2 → {2, False}, 3 → {3, False}, 4 → {4, False}, 5 → {5, False}|>
  , order={J, M, Z, Q, E, L, V, Y, X, T, I, G, P}
consider={H, N}

workers=
  <|1 → {1, True, H, 548}, 2 → {2, False}, 3 → {3, False}, 4 → {4, False}, 5 → {5, False}|>
consider={N}

workers=
  <|1 → {1, True, H, 548}, 2 → {2, True, N, 554}, 3 → {3, False}, 4 → {4, False}, 5 → {5, False}|>
sorted busy workers=<|1 → {1, True, H, 548}, 2 → {2, True, N, 554}|>

time=548, added=H, workers=
  <|1 → {1, False}, 2 → {2, True, N, 554}, 3 → {3, False}, 4 → {4, False}, 5 → {5, False}|>
  , order={J, M, Z, Q, E, L, V, Y, X, T, I, G, P, H}
consider={F, S}

workers=
  <|1 → {1, True, F, 614}, 2 → {2, True, N, 554}, 3 → {3, False}, 4 → {4, False}, 5 → {5, False}|>
consider={S}

workers=<|1 → {1, True, F, 614}, 2 → {2, True, N, 554},
  3 → {3, True, S, 627}, 4 → {4, False}, 5 → {5, False}|>
sorted busy workers=<|2 → {2, True, N, 554}, 1 → {1, True, F, 614}, 3 → {3, True, S, 627}|>

time=554, added=N, workers=
  <|1 → {1, True, F, 614}, 2 → {2, False}, 3 → {3, True, S, 627}, 4 → {4, False}, 5 → {5, False}|>
  , order={J, M, Z, Q, E, L, V, Y, X, T, I, G, P, H, N}
sorted busy workers=<|1 → {1, True, F, 614}, 3 → {3, True, S, 627}|>

time=614, added=F, workers=
  <|1 → {1, False}, 2 → {2, False}, 3 → {3, True, S, 627}, 4 → {4, False}, 5 → {5, False}|>
  , order={J, M, Z, Q, E, L, V, Y, X, T, I, G, P, H, N, F}
sorted busy workers=<|3 → {3, True, S, 627}|>

time=627, added=S, workers=
  <|1 → {1, False}, 2 → {2, False}, 3 → {3, False}, 4 → {4, False}, 5 → {5, False}|>
  , order={J, M, Z, Q, E, L, V, Y, X, T, I, G, P, H, N, F, S}
consider={0}

workers=
  <|1 → {1, True, 0, 702}, 2 → {2, False}, 3 → {3, False}, 4 → {4, False}, 5 → {5, False}|>
sorted busy workers=<|1 → {1, True, 0, 702}|>

time=702, added=0, workers=
  <|1 → {1, False}, 2 → {2, False}, 3 → {3, False}, 4 → {4, False}, 5 → {5, False}|>
  , order={J, M, Z, Q, E, L, V, Y, X, T, I, G, P, H, N, F, S, 0}
consider={A, D}

```



```

workers=
  <|1 → {1, True, A, 763}, 2 → {2, False}, 3 → {3, False}, 4 → {4, False}, 5 → {5, False}|>
consider={D}

workers=
  <|1 → {1, True, A, 763}, 2 → {2, True, D, 766}, 3 → {3, False}, 4 → {4, False}, 5 → {5, False}|>
sorted busy workers=<|1 → {1, True, A, 763}, 2 → {2, True, D, 766}|>
time=763, added=A, workers=
  <|1 → {1, False}, 2 → {2, True, D, 766}, 3 → {3, False}, 4 → {4, False}, 5 → {5, False}|>
  , order={J, M, Z, Q, E, L, V, Y, X, T, I, G, P, H, N, F, S, O, A}
sorted busy workers=<|2 → {2, True, D, 766}|>
time=766, added=D, workers=
  <|1 → {1, False}, 2 → {2, False}, 3 → {3, False}, 4 → {4, False}, 5 → {5, False}|>
  , order={J, M, Z, Q, E, L, V, Y, X, T, I, G, P, H, N, F, S, O, A, D}
consider={K, W}

workers=
  <|1 → {1, True, K, 837}, 2 → {2, False}, 3 → {3, False}, 4 → {4, False}, 5 → {5, False}|>
consider={W}

workers=
  <|1 → {1, True, K, 837}, 2 → {2, True, W, 849}, 3 → {3, False}, 4 → {4, False}, 5 → {5, False}|>
sorted busy workers=<|1 → {1, True, K, 837}, 2 → {2, True, W, 849}|>
time=837, added=K, workers=
  <|1 → {1, False}, 2 → {2, True, W, 849}, 3 → {3, False}, 4 → {4, False}, 5 → {5, False}|>
  , order={J, M, Z, Q, E, L, V, Y, X, T, I, G, P, H, N, F, S, O, A, D, K}
sorted busy workers=<|2 → {2, True, W, 849}|>
time=849, added=W, workers=
  <|1 → {1, False}, 2 → {2, False}, 3 → {3, False}, 4 → {4, False}, 5 → {5, False}|>
  , order={J, M, Z, Q, E, L, V, Y, X, T, I, G, P, H, N, F, S, O, A, D, K, W}
consider={B}

workers=
  <|1 → {1, True, B, 911}, 2 → {2, False}, 3 → {3, False}, 4 → {4, False}, 5 → {5, False}|>
sorted busy workers=<|1 → {1, True, B, 911}|>
time=911, added=B, workers=
  <|1 → {1, False}, 2 → {2, False}, 3 → {3, False}, 4 → {4, False}, 5 → {5, False}|>
  , order={J, M, Z, Q, E, L, V, Y, X, T, I, G, P, H, N, F, S, O, A, D, K, W, B}
consider={R}

workers=
  <|1 → {1, True, R, 989}, 2 → {2, False}, 3 → {3, False}, 4 → {4, False}, 5 → {5, False}|>
sorted busy workers=<|1 → {1, True, R, 989}|>

```

```

time=989, added=R, workers=
  <|1 → {1, False}, 2 → {2, False}, 3 → {3, False}, 4 → {4, False}, 5 → {5, False}|>
  , order={J, M, Z, Q, E, L, V, Y, X, T, I, G, P, H, N, F, S, O, A, D, K, W, B, R}
consider={U}
workers=
  <|1 → {1, True, U, 1070}, 2 → {2, False}, 3 → {3, False}, 4 → {4, False}, 5 → {5, False}|>
sorted busy workers=<|1 → {1, True, U, 1070}|>
time=1070, added=U, workers=
  <|1 → {1, False}, 2 → {2, False}, 3 → {3, False}, 4 → {4, False}, 5 → {5, False}|>
  , order={J, M, Z, Q, E, L, V, Y, X, T, I, G, P, H, N, F, S, O, A, D, K, W, B, R, U}
consider={C}
workers=
  <|1 → {1, True, C, 1133}, 2 → {2, False}, 3 → {3, False}, 4 → {4, False}, 5 → {5, False}|>
sorted busy workers=<|1 → {1, True, C, 1133}|>
time=1133, added=C, workers=
  <|1 → {1, False}, 2 → {2, False}, 3 → {3, False}, 4 → {4, False}, 5 → {5, False}|>
  , order={J, M, Z, Q, E, L, V, Y, X, T, I, G, P, H, N, F, S, O, A, D, K, W, B, R, U, C}
Out[=]= {JMZQELVYXTIGPHNFSOADKWBRUC, 1133}

```