Hiring Janitors

Dean Capellari, 2019

Introduction

- A museum has open job spots for janitors and is currently hiring. They would like to know:
    - How many janitors they need to hire.
    - A work schedule for the janitors.
    - An algorithm to find both using the number of guests.

Problem Statement

- Everyday, x guests come to the museum, where $1000 \leq x \leq 1500$.
- Guests produce $99x/100$ total trash per day.
- One janitor can clean 100 trash in 1 hour; the maximum work day is 4 hours per day, and the maximum work week is 6 days.
- The max number of janitors is 25, and the minimum number is 1.
- Find an algorithm using the number of people and the work hours to find the proper number of janitors required to clean the trash.

Results p1

- First results:
  - The first equation I found was $j = \left(\frac{\left(\frac{99x}{100}\right)}{th}\right)$
  - List of variables:
    - h=hours a day
    - j=#of janitors.
    - x=#of guests.
    - a=hrs on the job so far.
    - y=total janitors.
    - l is janitors off.
    - t=#of trash cleanable in 1 hour(important for fatigue),t=100(for now).
    - z=extra people coming in every week(important for ad campaign)

VARIABLES:
h=hours a day. j=#of janitors.
x=#of guests. a=hrs on the job so far.
y=total janitors. l is janitors off.
t=#of trash cleanable in 1 hour.
z=extra people coming in every week.
a=hours previously worked that day.

Results p2

- First results (cont.):
  - Schedule Table to the right:
  - The general algorithm for the table:
    - hire at minimum twice the number of janitors for day 1
    - for the next day take j janitors from off duty to on duty
    - the janitors on duty the previous day get the day off
  - In the table, you have janitors 1, 2, 3, and 4 working, and 5 thru 11 resting. For day two, you have 5, 6, 7, and 8 on the job, and 9, 10, 11, 1, 2, 3, and 4 resting.

y=11

	Work(j)	off(l)
s	1 2 3 4	5 6 7 8 9 10 11
m	5 6 7 8	9 10 11 1 2 3 4
t	9 10 11 1	2 3 4 5 6 7 8
w	2 3 4 5	6 7 8 9 10 11 1
t	6 7 8 9	10 11 1 2 3 4 5
f	10 11 1 2	3 4 5 6 7 8 9
s	3 4 5 6	7 8 9 10 11 1 2

VARIABLES:
h=hours a day. j=#of janitors.
x=#of guests. a=hrs on the job so far.
y=total janitors. l is janitors off.
t=#of trash cleanable in 1 hour.
z=extra people coming in every week.
a=hours previously worked that day.

Results p3

- Ad Campaign:
  - The museum starts an ad campaign that brings in *Z* more people a week.
  - Equation: $X_n = z + X_{(n-1)}$
  - plug into original equation.
  - Make Table.

VARIABLES:
h=hours a day. j=#of janitors.
x=#of guests. a=hrs on the job so far.
y=total janitors. l is janitors off.
t=#of trash cleanable in 1 hour.
z=extra people coming in every week.
a=hours previously worked that day.

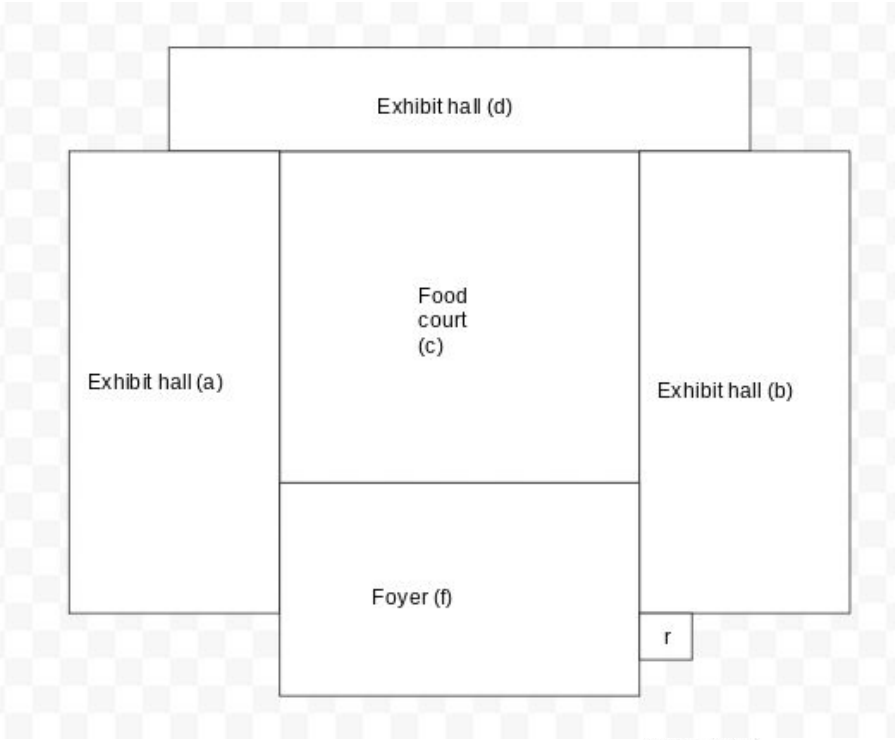
Results p4

- Fatigue:
  - Fatigue Equation: $t=100-10a$
    - t=cleanable trash per hour for 1 janitor
    - a=number of hours previously worked that day
  - For hour 1:
    - $a=0$, $t=100$
  - For hour 2:
    - $a=1$, $t=90$
  - And so on.
  - Plug into original equation and then into table algorithm.

VARIABLES:
h=hours a day. j=#of janitors.
x=#of guests. a=hrs on the job so far.
y=total janitors. l is janitors off.
t=#of trash cleanable in 1 hour.
z=extra people coming in every week.
a=hours previously worked that day.

Results p5

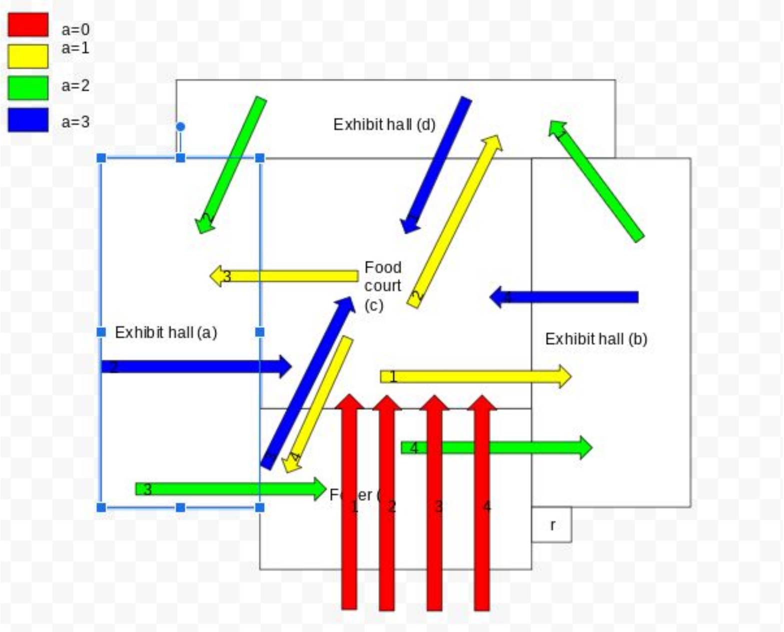
- Path Algorithm p1:
  - The museum wants us to find a path for janitors so that all trash is cleaned and all previously mentioned constraints are followed.
  - This is where we lose a single definitive algorithm.



VARIABLES:
h=hours a day. j=#of janitors.
x=#of guests. a=hrs on the job so far.
y=total janitors. l is janitors off.
t=#of trash cleanable in 1 hour.
z=extra people coming in every week.
a=hours previously worked that day.

Results p6

- Path Algorithm p2
  - Best Algorithm for j=4, h=4, and x=1000:
    - a=0: Send everyone to c.
    - a=1: one to a, one to b, one to d, one to f.
    - a=2:one to a, one to b, one to d, one to f.
    - a=3: all to c.
    - When done go to r.



VARIABLES:
h=hours a day. j=#of janitors.
x=#of guests. a=hrs on the job so far.
y=total janitors. l is janitors off.
t=#of trash cleanable in 1 hour.
z=extra people coming in every week.
a=hours previously worked that day.

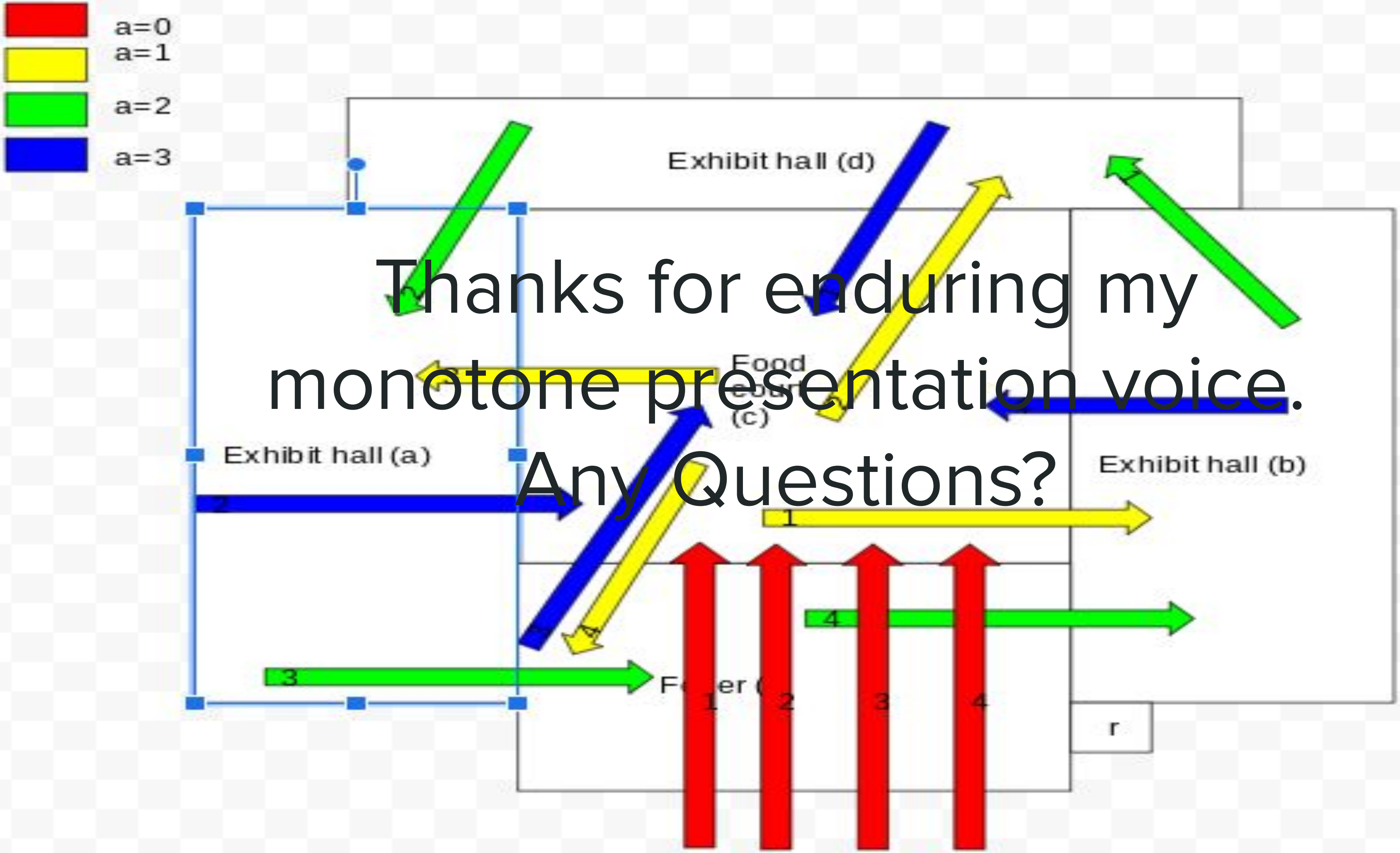
Results p7

- Path Algorithm p3:
  - This general process is applicable for all $1 \leq y \leq 25$:
    - Send all janitors to dirtiest room.
    - Send as close to an equal number of janitors to each room as you can get.
    - If any trash left in dirtiest room and sending all janitors to dirty room during last hour does not work, send enough janitors to dirty room as needed during step 2.
    - As stated, and if needed, send all janitors to dirtiest room.
    - If finished go to breakroom.

VARIABLES:
h=hours a day. j=#of janitors.
x=#of guests. a=hrs on the job so far.
y=total janitors. l is janitors off.
t=#of trash cleanable in 1 hour.
z=extra people coming in every week.
a=hours previously worked that day.

Conclusion

- It is solvable. Everything up to the path algorithm was very straightforward, but finding paths took much longer than expected.
- What I would do next time:
  - I would probably add in the restriction that janitors can only walk into rooms from adjacent rooms.



Thanks for enduring my monotone presentation voice.

Any Questions?