Sanitisation Routine

Sujit Kumar Chakrabarti

July 14, 2020

1 Introduction

During pandemic, returning home from outside errands is always an irritating moment. You are supposed to go through this entire process of sanitisation which is so messy! In principle, during the process whatever comes in physical contact with anything supposedly contaminated also gets contaminated. Sometimes, despite observing diligence and care, we end up feeling that we have ended up spreading the virus all over the house. If we try to be too careful, it appears that we are never going to make it through the damn process; whole bottles of sanitisers are going to go empty; and our hands are will be skinned off due to repeated soap water washing.

In this article, we will try to develop some understanding about the real process of sanitisation. No! No hi-fi stuff! Plain common sense, may be with a dash of mathematics to help us tell things succinctly. We will present a sytematic approach to sanitisation, good enough at least to give us a confidence that sanitisation can pracically be got over with without emptying bottles of sanitisers or peeling the skin off our hands.

So, how to think about sanitisation? To start with, we have:

1. Contaminated/dirty/infected objects
2. Sanitised/clean/uninfected objects
3. Our hand

There are two types of actions, primarily:

1. Sanitisation/washing
2. Others

We can also start with a bunch of assumptions:

1. If anything is cleaned, it is clean afterwards regardless of its initial state.
2. If any object X is involved in any action (other than sanitisation) with any other object Y which is dirty, then X is dirty afterwards.

2 Precedence order between Actions

A precedence order is assumed to be defined among the actions: $A \rightarrow B$ means that A happens before B. The ordering dependencies naturally fit into a directed acyclic graph model due to their partially ordered nature. Any topological sorting of this DAG is a minimally feasible schedule. The key optimisation step would be to choose appropriate points in this schedule and appropriate objects to sanitise so that the overall cost of sanitisation can be kept to the minimum.

Example

Objects

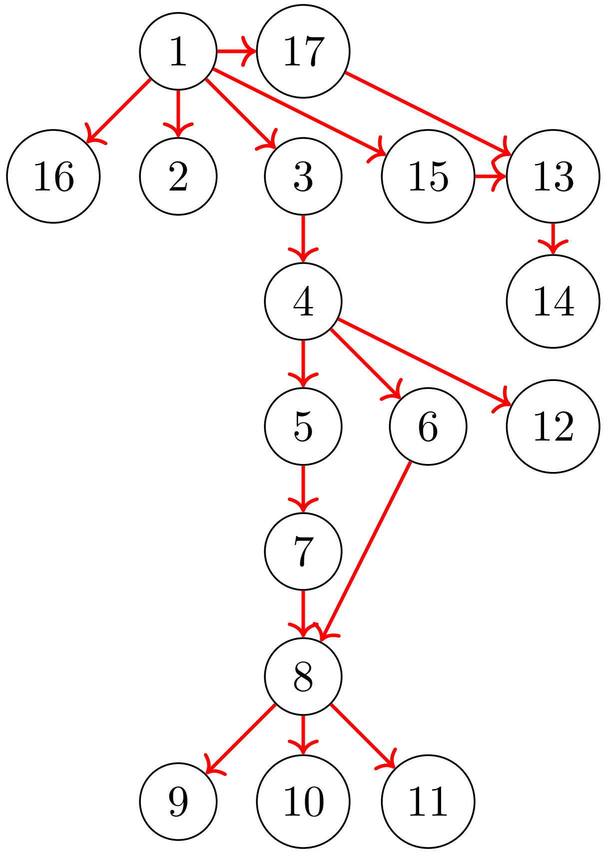
Name	Initial	Final
Door	$\times$	$\times$
Key	$\times$	$\checkmark$
Milk packet	$\times$	$\checkmark$
Sink	$\times$	$\times$
Scissors	$\checkmark$	$\checkmark$
Container	$\checkmark$	$\checkmark$
Bag	$\times$	$\times$
Outside clothes	$\times$	$\times$
Home clothes	$\checkmark$	$\checkmark$
Phone	$\times$	$\checkmark$
Mask	$\times$	$\times$
Purse	$\times$	$\times$

Above a $\checkmark$represents clean and a $\times$ represents dirty.

Actions

Code	Name
1	Open door
2	Keep key
3	Take out milk packet from bag
4	Keep milk packet in sink
5	Take scissors
6	Take container
7	Cut packet with scissors
8	Pour milk from packet to container
9	Replace scissors
10	Put container
11	Dispose packet
12	Replace bag
13	Take off outside clothes
14	Put on home clothes
15	Take out phone
16	Take out mask
17	Take out purse

Dependencies



3 Cost of Sanitisation

In certain cases sanitisation is unavoidable. For example, an object which needs to be clean at the end of the process must be sanitised at least once. However, in certain other cases, sanitisation happens as an unavoidable overhead. Such sanitisation is of the following types:

1. Sanitising the hand alone.
2. Sanitising a contaminated object which otherwise doesn't need to be sanitised per se (e.g. because it will be eventually disposed), but must be sanitised because it will be part of some action involving other clean objects.
3. Sanitising an otherwise clean object because it came in contact with a contaminated object during the sanitisation routine.

The actual cost of clean/sanitising an object is not uniform across objects. Some are easy to clean (e.g. small items like keys), some are harder (e.g. vegetables), while there are others which are very difficult or practically impossible to clean, e.g. big packets or large clothing items.

4 Objectives

The objective of a sanitisation process is to not contaminate any clean object, including our hands. Some of the contaminated objects must be sanitised while some needn't be bothered about. The main objective is to optimise the process in terms of the cost incurred due to sanitisation.

Viewed as a scheduling problem, we want to come up with a schedule S, which is nothing but a sequence of actions such that:

1. All actions are done.

2. All objects required to be clean at the end of the process should indeed be clean.

3. There should be no other schedule S' which has lower cost than S.

5 Scheduling Algorithms

In this section, we present the various algorithms we design and implement to solve the problem presented in the above sections.

We view the scheduling problem as having the following two components

1. *Generation of skeletal sequence*, wherein the dependence graph is used to generate a sequence of essential actions.

2. *Introduction of sanitisation*, wherein sanitisation actions are inserted to achieve the scheduling objectives in an optimal way.

5.1 Generation of skeletal sequence

The first step is a variant of the well-known topological sort algorithm which runs on the dependency graph and generates a valid total ordering of the essential actions. That algorithm maintains a ready list of actions, i.e. actions which have none of the dependence actions (from the DAG) pending. Out of this ready list the algorithm picks up one which it deems the best to be dispatched next based on a heuristic. This is the step where this algorithm differs from the traditional topological sort algorithm.

```
procedure SKELETAN-SEQUENCE(D : DAG)
    schedule ← []
    ready ← [D.root]
    while ready ≠ [] do
        a ← BEST-ACTION(ready)
        Pop a from ready
        dep ← a.succ                    ▷ a.succ : the list of a's successors
        Add all elements of dep to ready
        Add a to schedule
```

6 Introduction of Sanitisation

6.1 Branch and Bound Algorithm

6.2 Genetic Algorithms

7 To Do

1. Complexity analysis of the problem