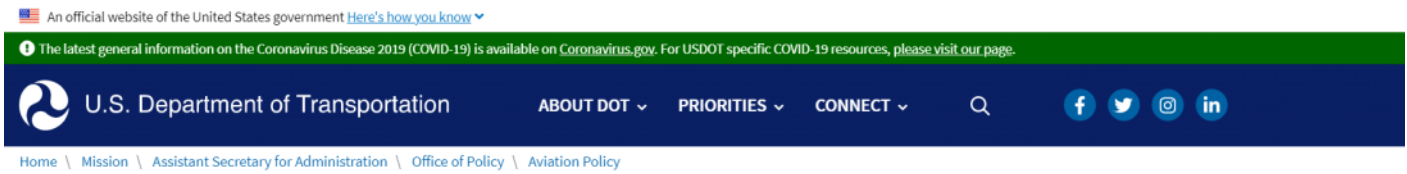
SCASDP



- Aviation Policy
- Licensing
- Small Community and Rural Air Service
  - Essential Air Service
  - Small Community Air Service Development Program (SCASDP)
  - Rural Airports
  - Alaskan Mail Rates
- Competition, Data, and Analysis
- International Relations
- Airline Rules and Fares
- RAISE Award Challenge Competition
- Who We Are

Small Community Air Service Development Program (SCASDP)

UPDATE:

As of January 19, 2021, the Department issued an Amendment to the November 24, 2020, NOFO. Please reference both FY 2019 Orders 2020-11-5 and 2021-1-4 (linked here) for full application details and updates. The FY 2019 NOFO Amendment is in response to the December 27, 2020, Coronavirus Response and Relief Supplemental Appropriations Act.

FY 2019 SCASDP Amended NOFO Issued: January 19, 2021

FY 2019 SCASDP NOFO (solicitation order) Issued : November 24, 2020

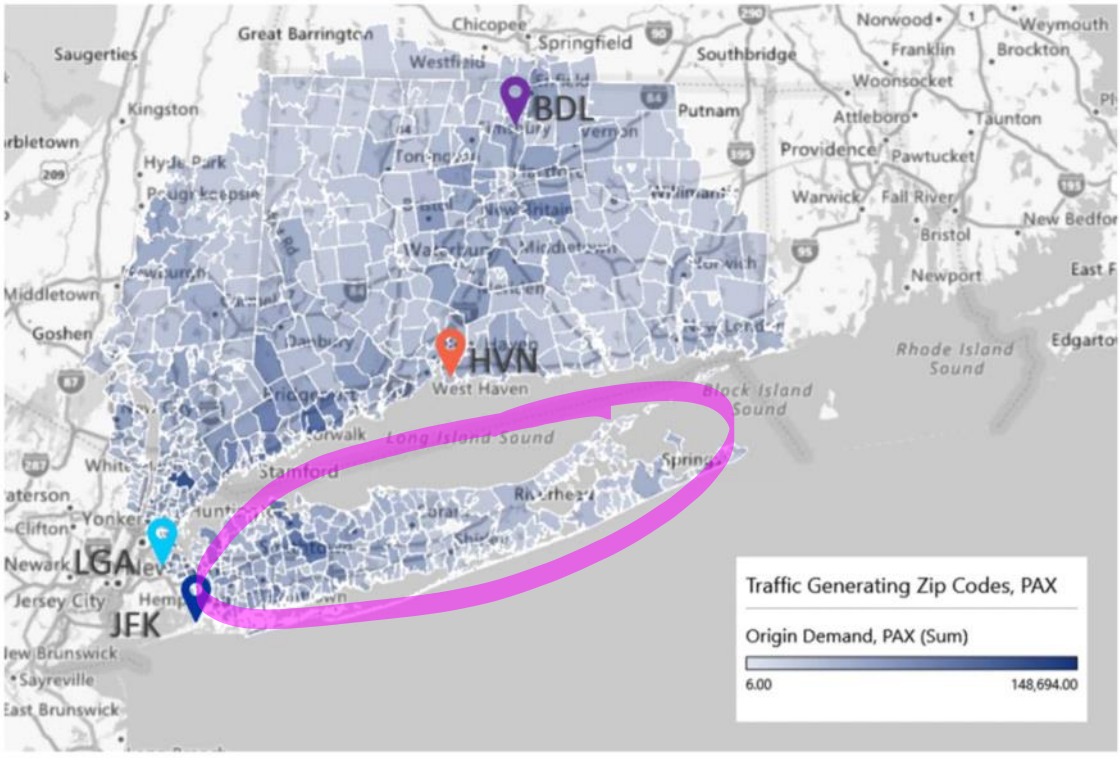
FY 2019 SCASDP SUMMARY FORM (*Fillable PDF version)

Overview

The Small Community Air Service Development Program (SCASDP), a grant program designed to help small communities address air service and airfare issues, is managed by the Associate Director, Small Community Air Service Development, under the Office of Aviation Analysis. The Essential Air Service (EAS) and SCASDP programs are separate and unique. SCASDP's eligibility criteria are broader and provide a grant applicant the opportunity to self-identify its air service deficiencies and propose an appropriate solution. EAS is a direct subsidy to air carriers and serves a limited universe of eligible applicants. Please refer to the EAS website for additional information. SCASDP can involve, among others, revenue guarantees, financial assistance for marketing programs, start-up costs and studies.

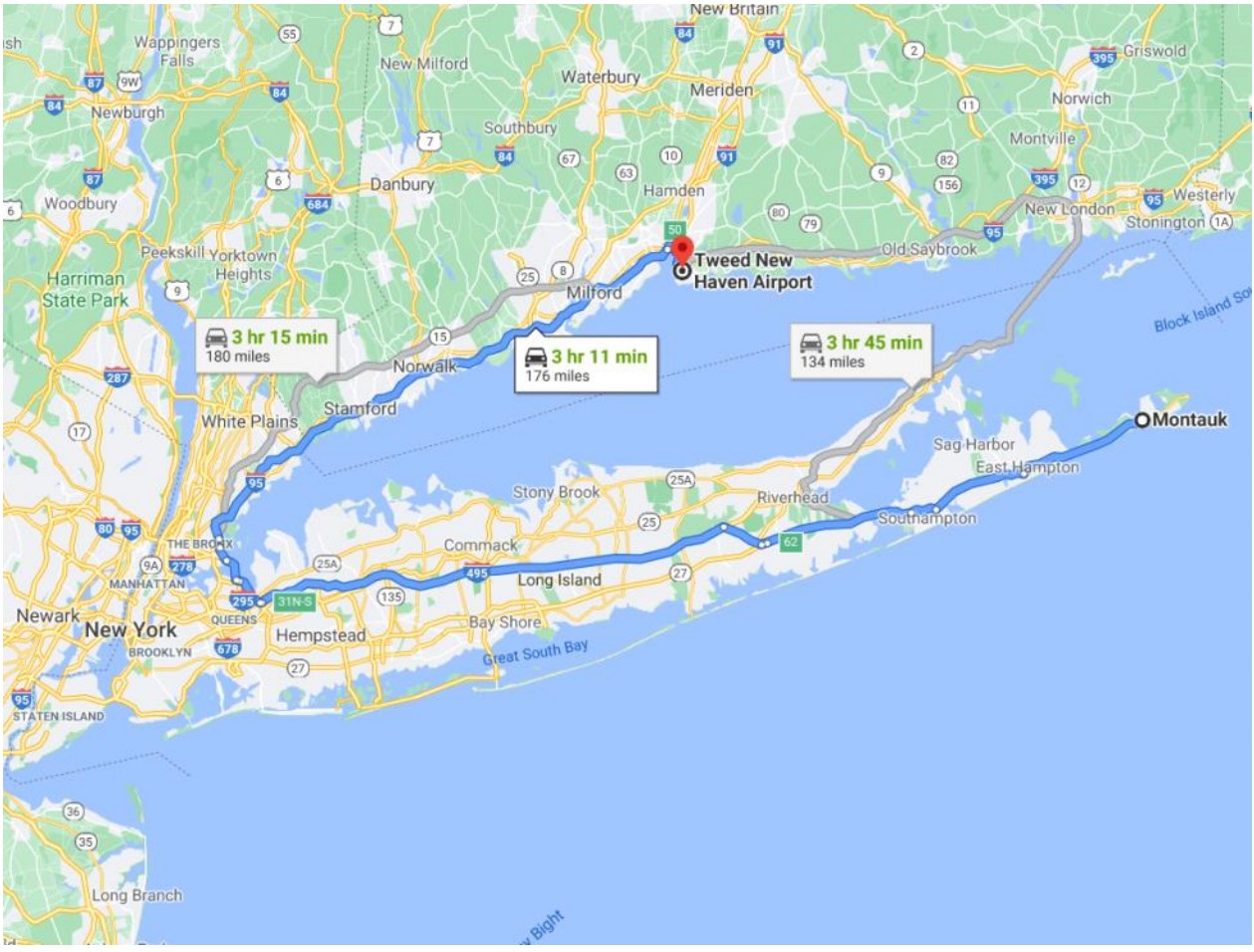
60-mile Radius

Figure 7. HVN's Catchment Traffic Generating Zip Codes within 60 mi of HVN (Resident/Origin Demand)

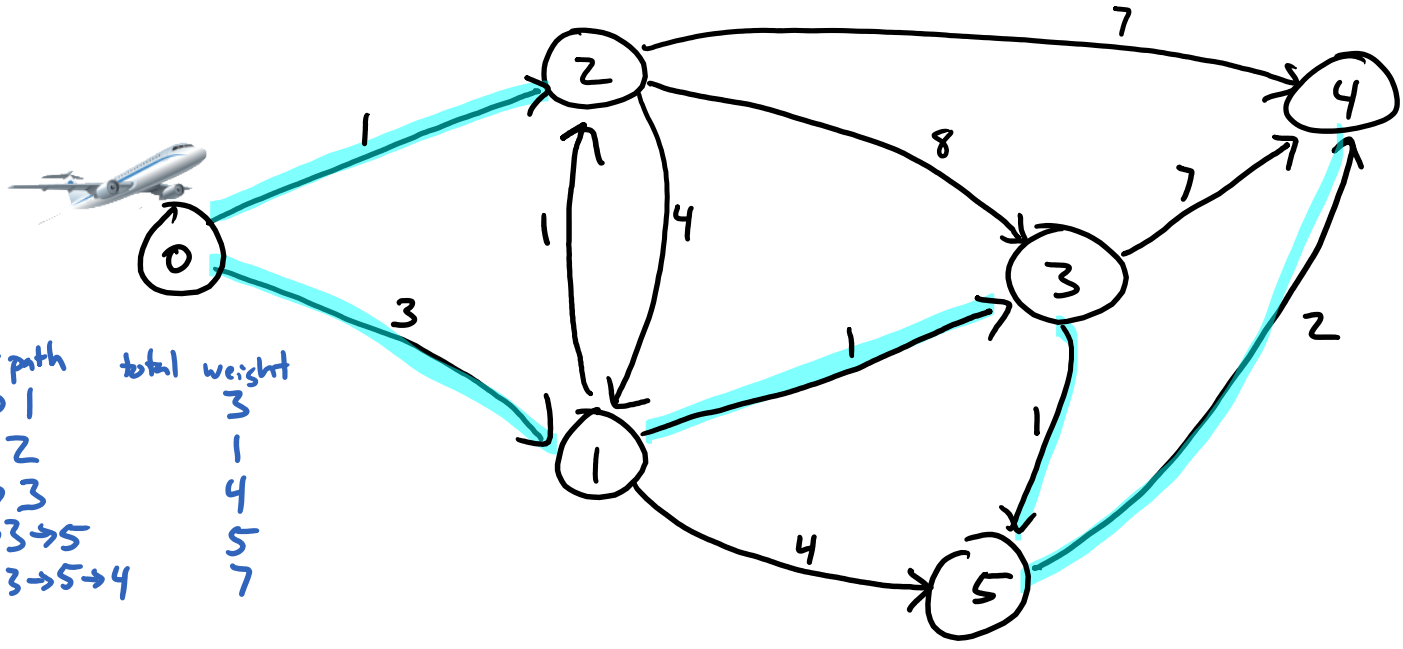


Source: ASM's HVN Catchment/Leakage Analysis, YE1Q2018, Airline Data Inc.

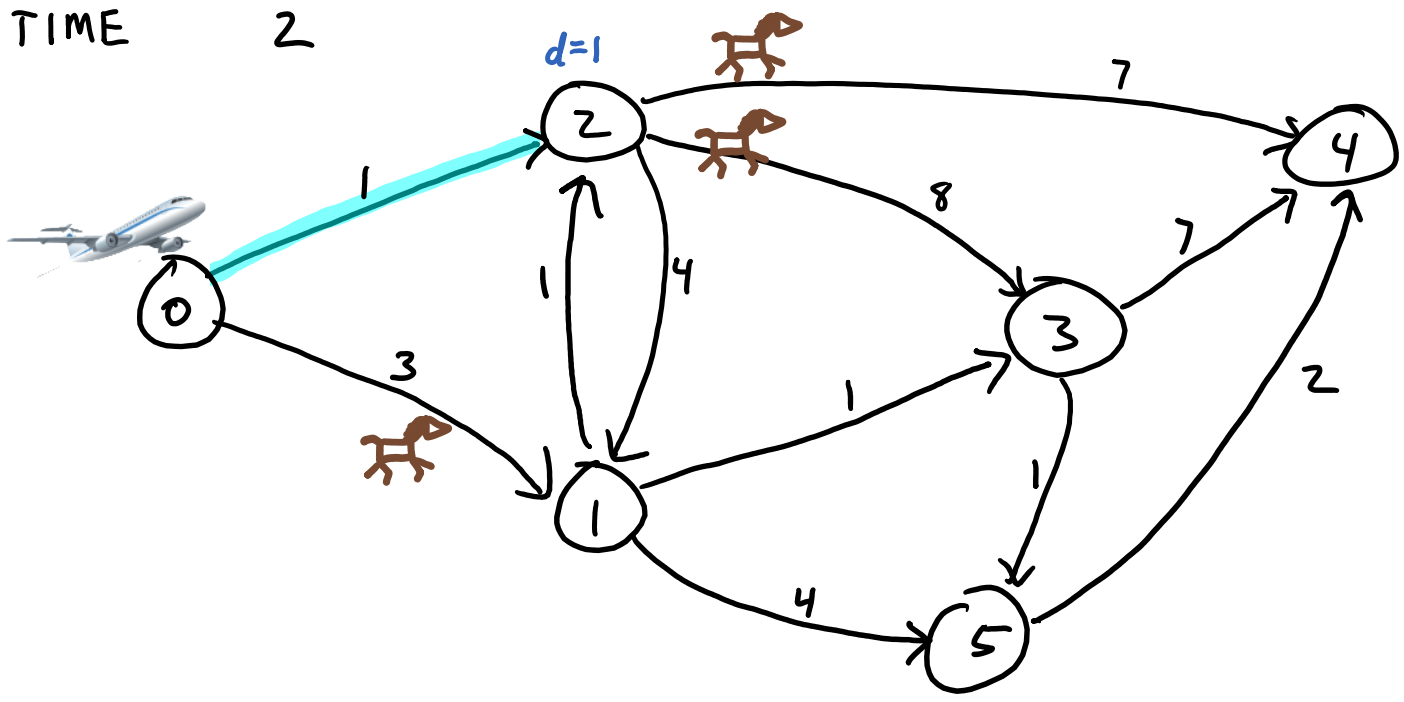
Long Islang is a long drive from HVN



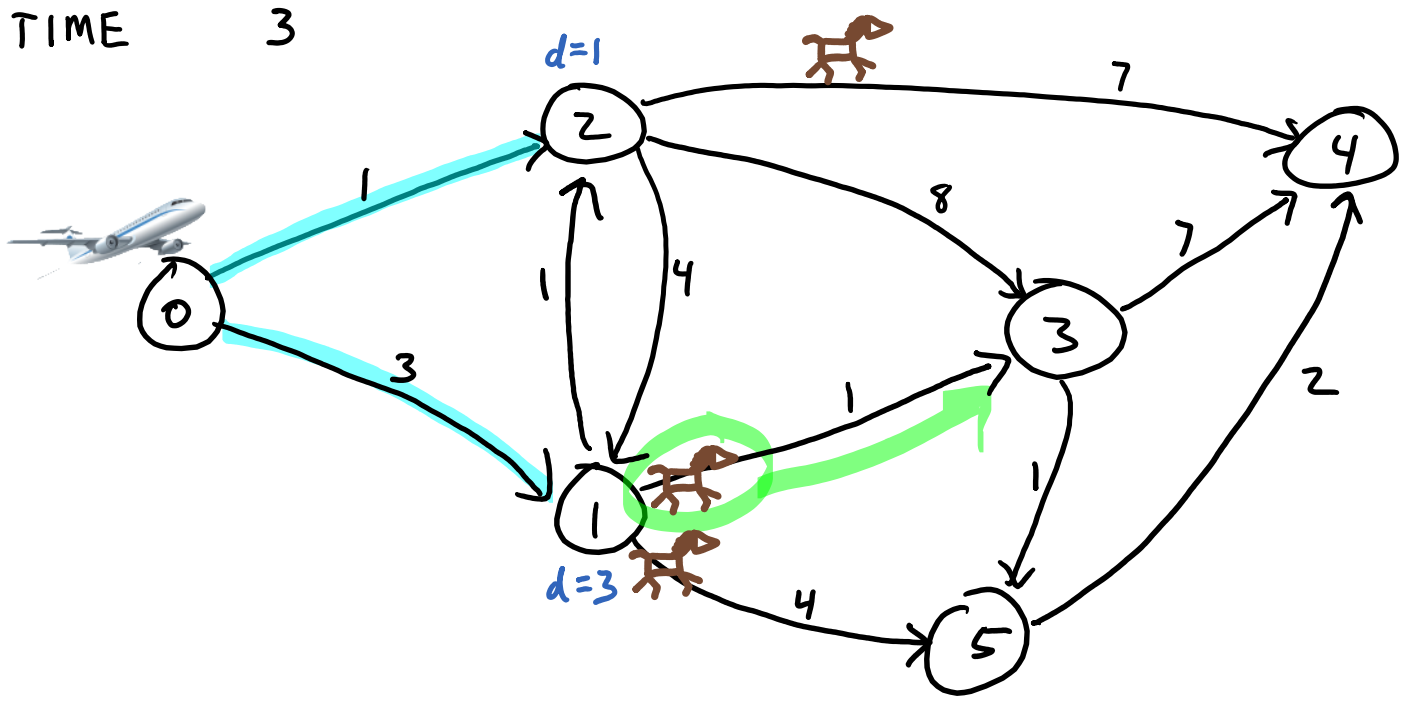
Shortest Paths



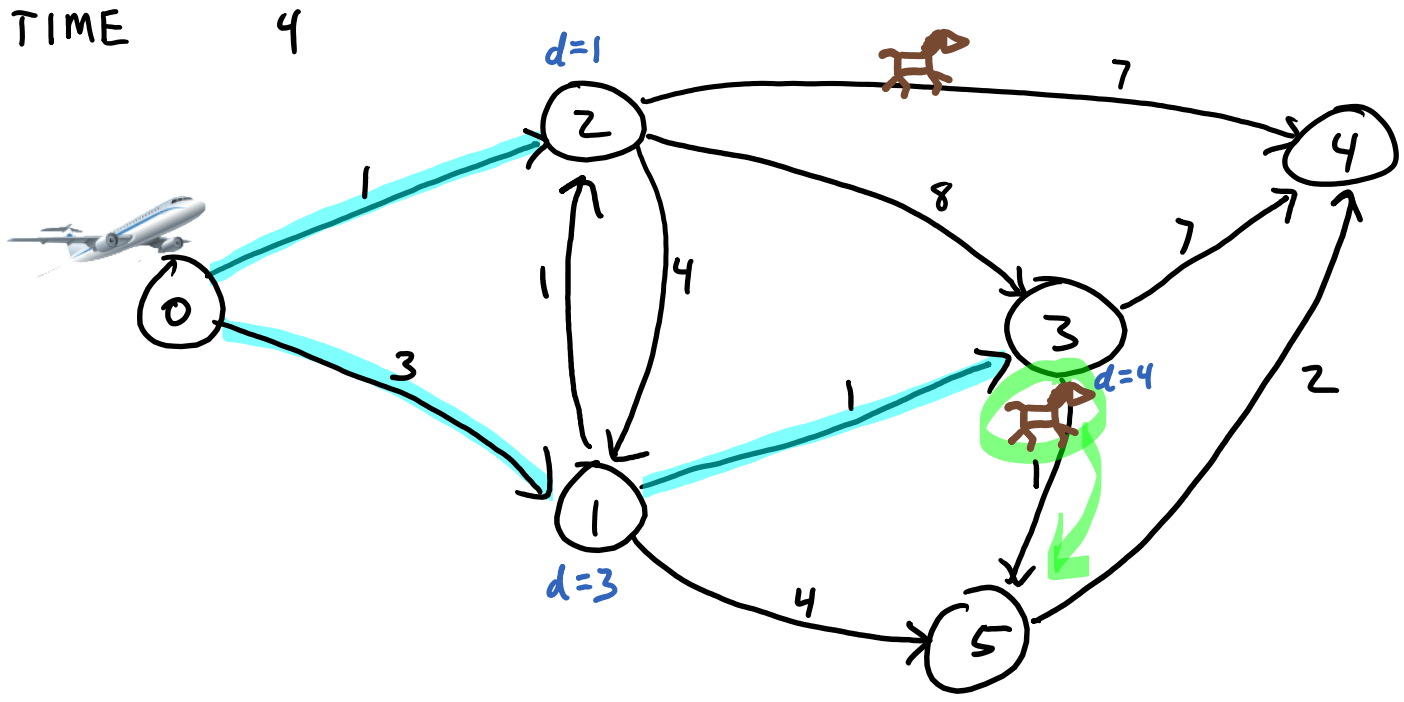
Shortest path	total weight
0 → 1	3
0 → 2	1
0 → 1 → 3	4
0 → 1 → 3 → 5	5
0 → 1 → 3 → 5 → 4	7


TIME 2
d=1
0
1
2
3
4
5
1
3
1
4
8
7
7
1
4
2

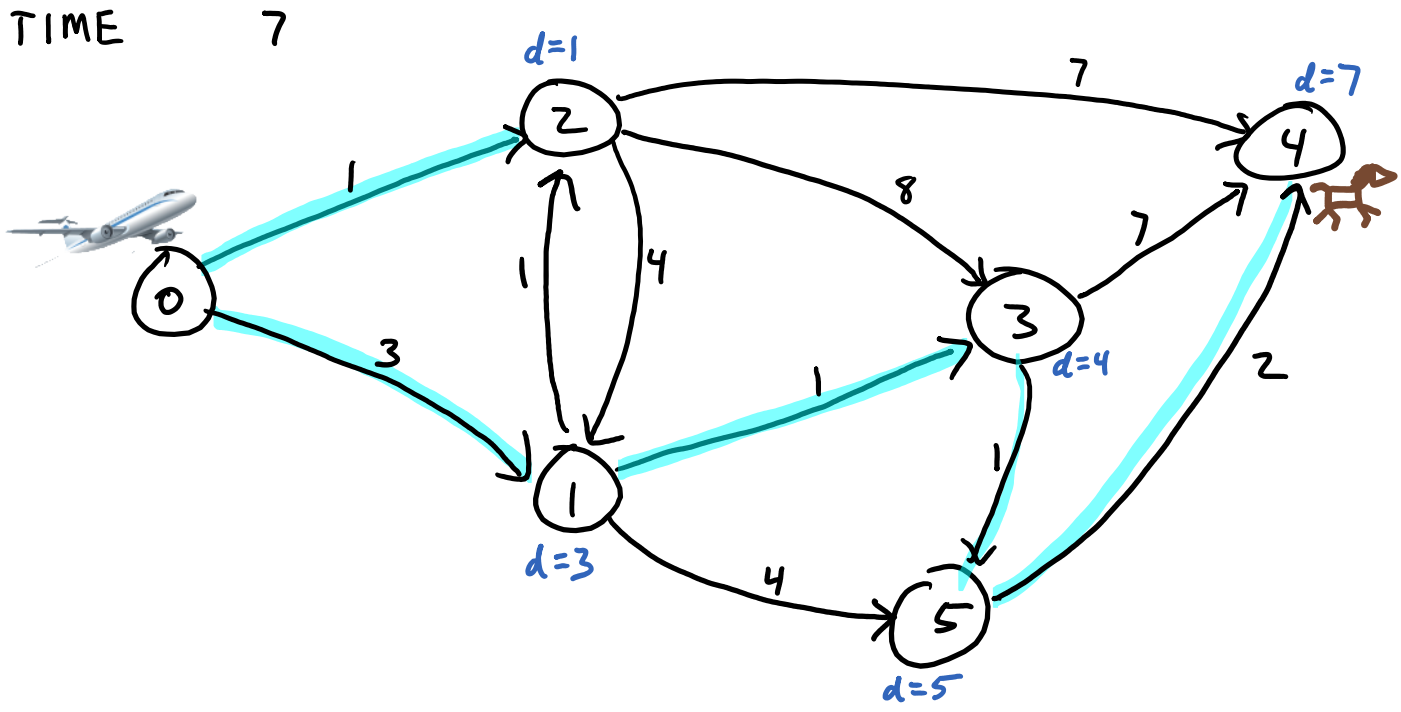
Shortest Paths



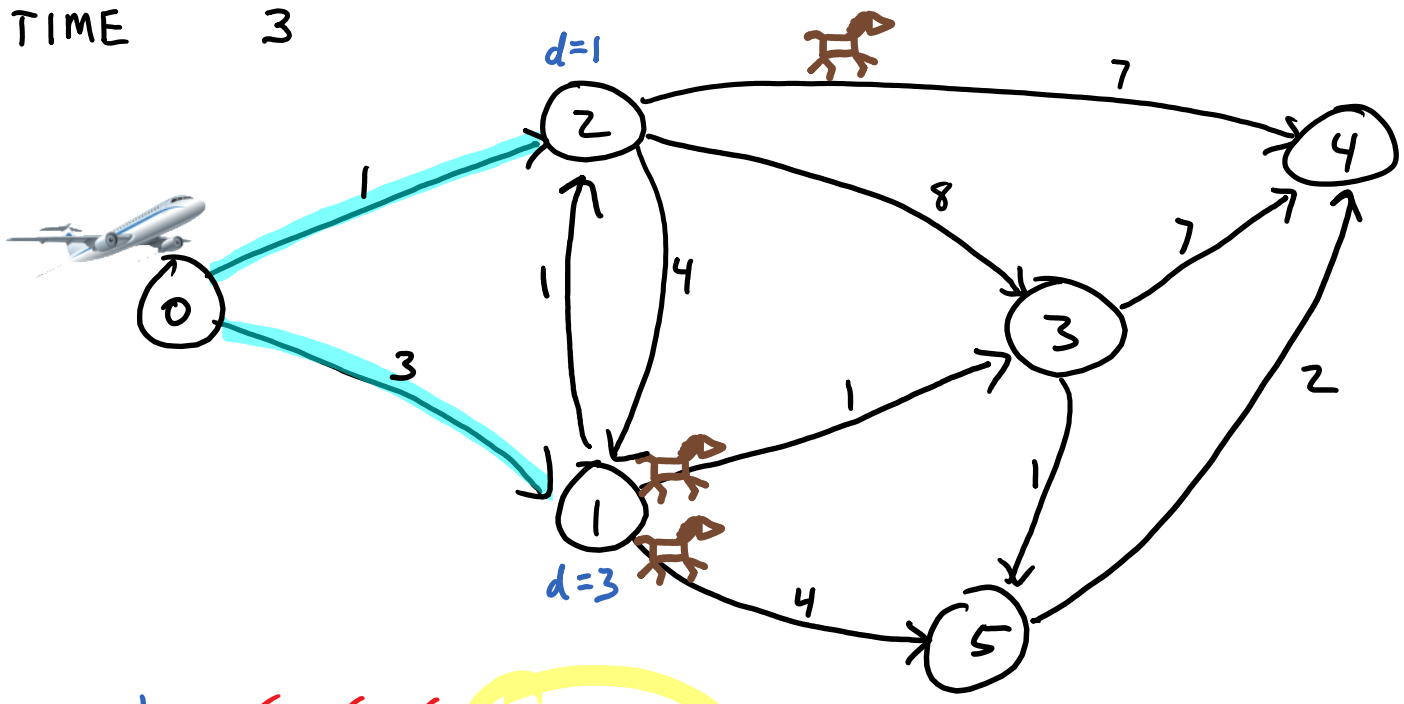
Shortest Paths



Shortest Paths

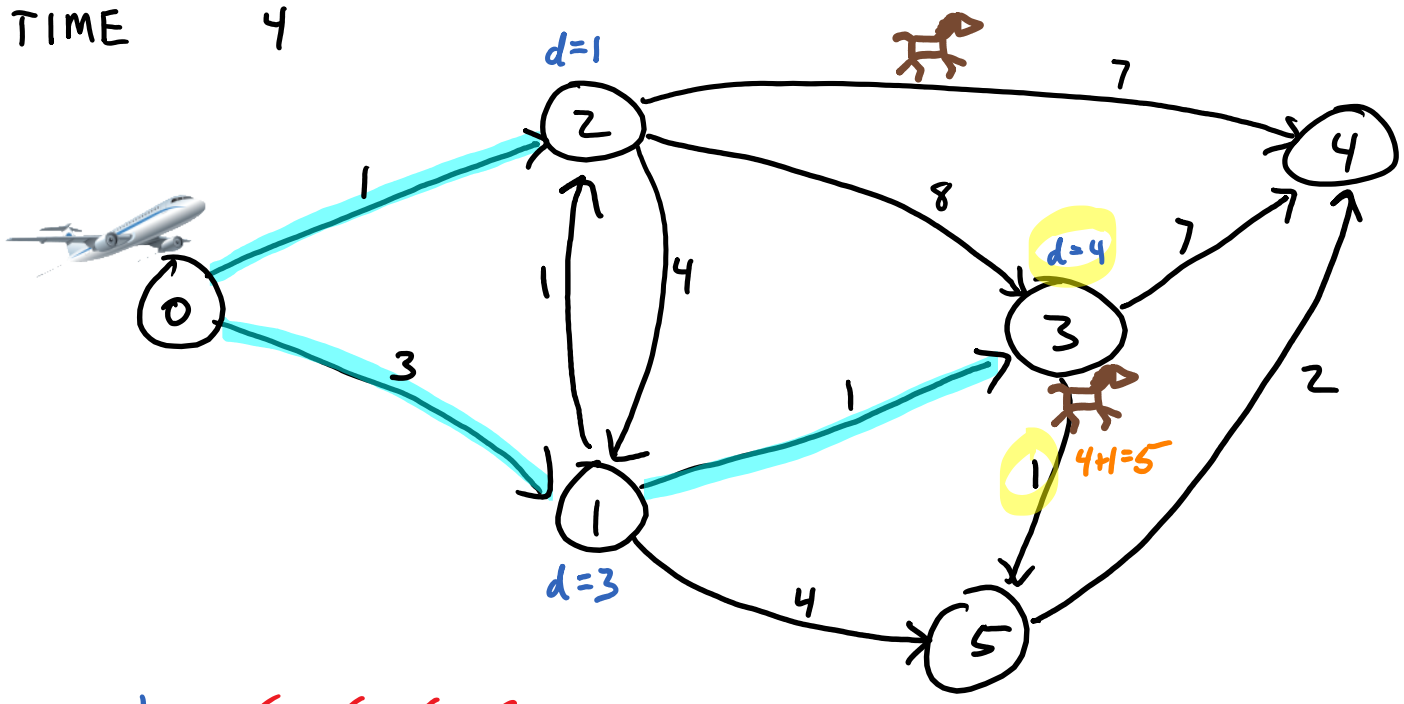


Shortest Paths



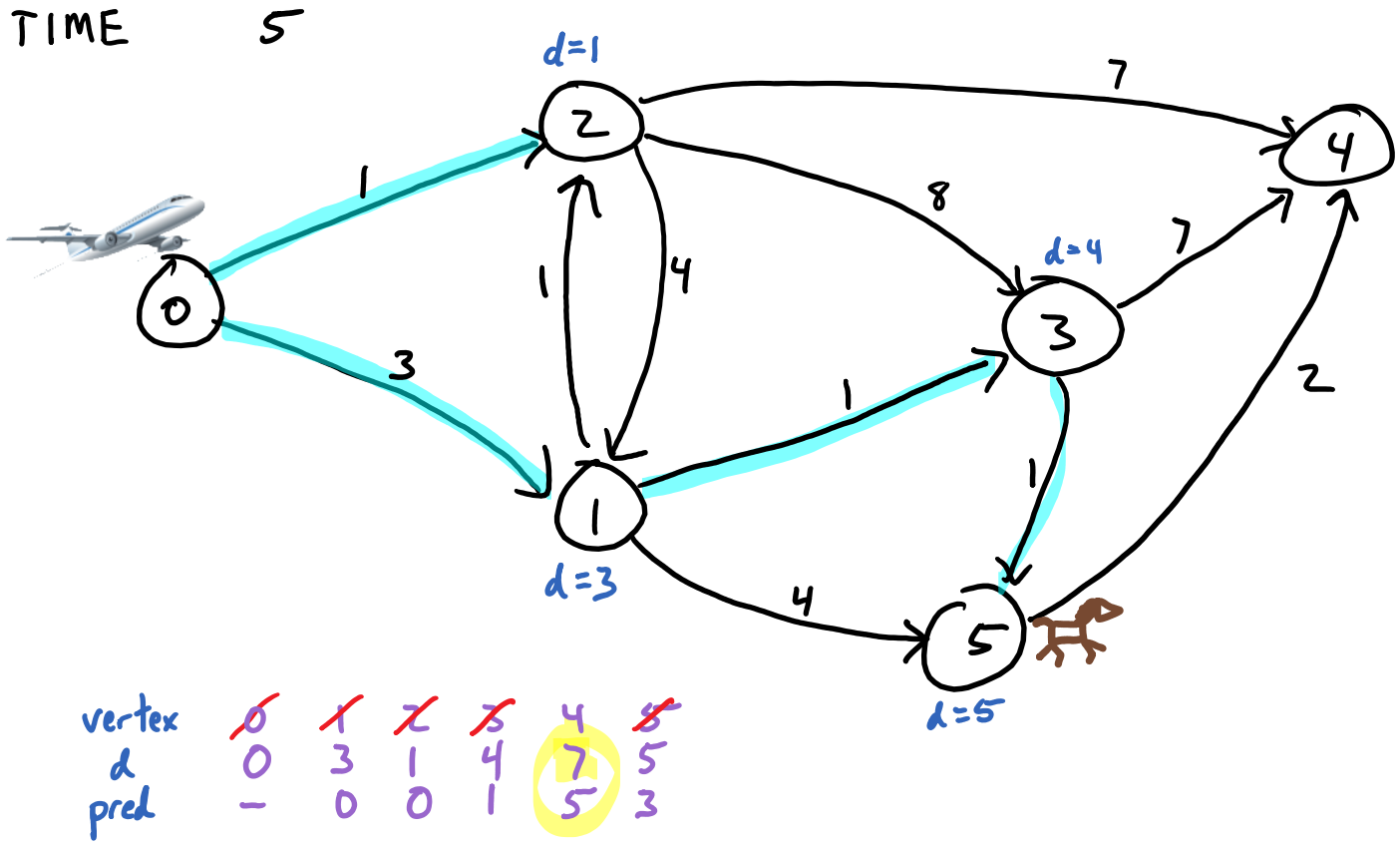
vertex	~~0~~	~~1~~	~~2~~	3	4	5
d	0	3	1	4	8	7
pred	–	0	0	1	2	1

Shortest Paths



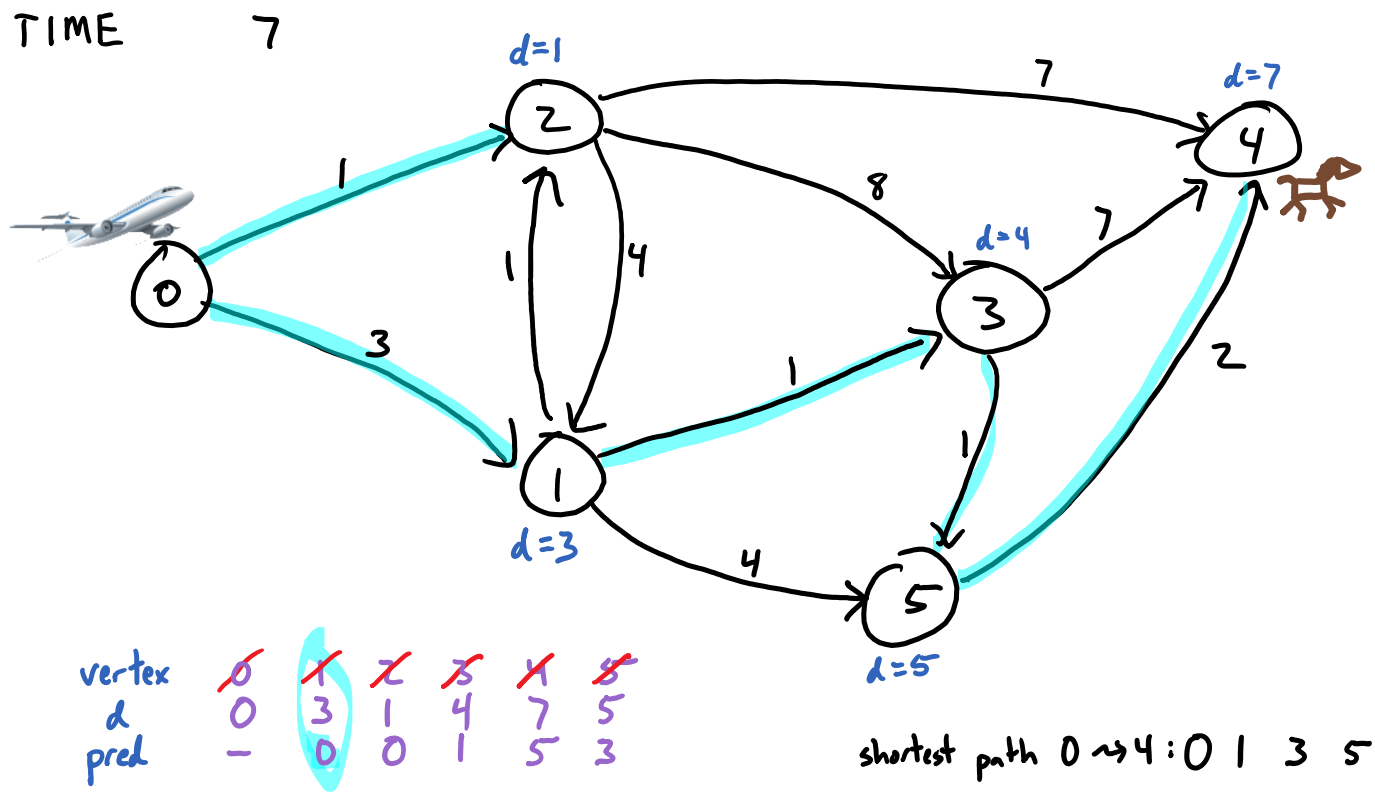
vertex	~~0~~	~~1~~	~~2~~	~~3~~	4	5
d	0	3	1	4	8	5
pred	–	0	0	1	2	3

Shortest Paths

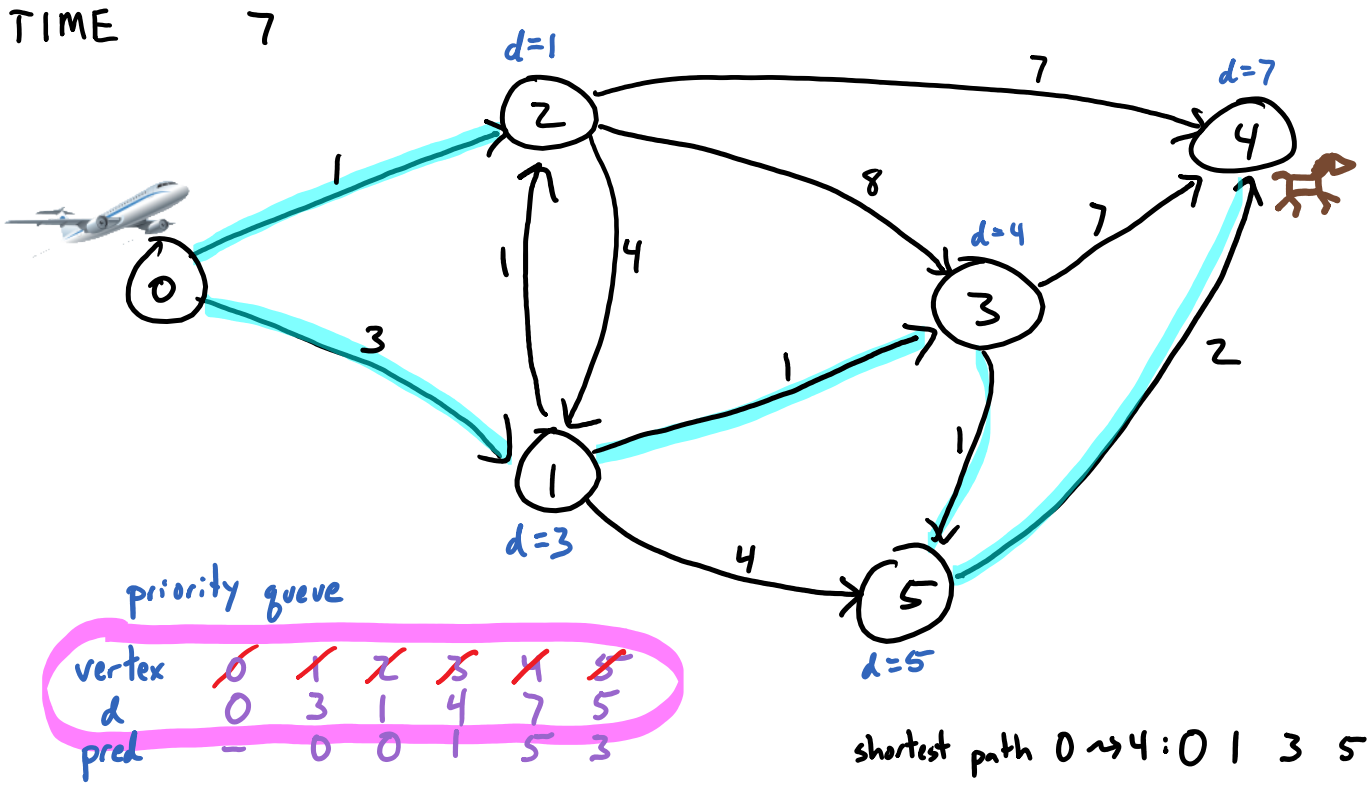


vertex	~~0~~	~~1~~	~~2~~	~~3~~	4	~~5~~
d	0	3	1	4	7	5
pred	–	0	0	1	5	3

Shortest Paths



Shortest Paths



Dijkstra's Algorithm

PRE: no negative weight edges

POST: d gives total weight of shortest paths, pred gives edges in shortest paths (∞ to mean unreachable)

```
for each v
  color[v], pred[v], d[v] ← IN_QUEUE, NIL, ∞
d[s] ← 0

Q ← new PriorityQueue(d)

while Q not empty
  u ← dequeue(Q)
  for each outneighbor v of u
    if color[v] = IN_QUEUE and d[v] > d[u] + w(u, v)
      change_priority(Q, v, d[u] + w(u, v))
      d[v] ← d[u] + w(u, v)
      pred[v] ← u
  color[u] ← DONE
```

one iteration per vertex

n calls

one iteration per edge

$\in m$ calls

adjacency list: $\Theta(n+m)$ + time for priority queue operations

Dijkstra's Algorithm

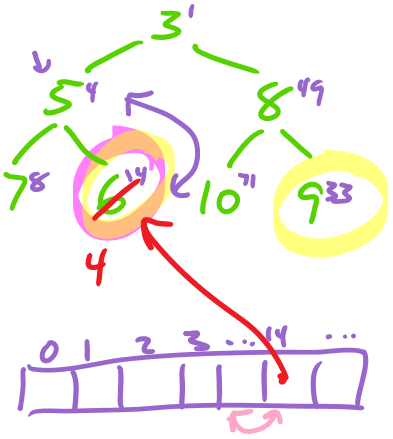
PRE: no negative weight edges

POST: d gives total weight of shortest paths, pred gives edges in shortest paths (∞ to mean unreachable)

```
for each v
  color[v], pred[v], d[v] ← IN_QUEUE, NIL, ∞
d[s] ← 0

Q ← new PriorityQueue(d)

while Q not empty                                   one iteration per vertex
  u ← dequeue(Q)                                    n calls  O(log n) each  total O(n log n)
  for each outneighbor v of u                       one iteration per edge
    if color[v] = IN_QUEUE and d[v] > d[u] + w(u, v)
      change_priority(Q, v, d[u] + w(u, v))         ∈ m calls  O(log n) each  O(m log n) total
      d[v] ← d[u] + w(u, v)
      pred[v] ← u
  color[u] ← DONE
```



adjacency list: $O(n+m)$ + time for priority queue operations

with heap: $O(n+m) + O(n \log n) + O(m \log n) = O(m \log n)$ (if all reachable so $m \geq n-1$)

with unsorted array: $O(n+m) + O(n^2) + O(m) = O(n^2)$

	heap	unsorted array
$m \in \Theta(n)$	$O(n \log n)$	$O(n^2)$
$m \in \Theta(n^2)$	$O(n^2 \log n)$	$O(n^2)$ ←

Tuesday, December 6, 2022 2:38 PM