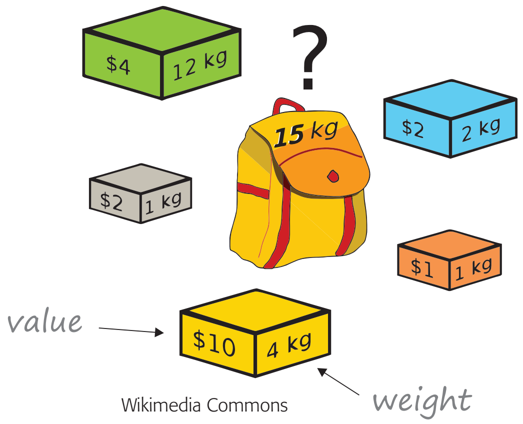
The Knapsack Problem



Wikimedia Commons

$$W, \{w_i\}, \{v_i\}$$

$$max \underset{\text{subset of } i}{\textstyle\sum_j} v_j \mid \textstyle\sum_j w_j \le W$$

A knapsack in the real world can be a shipping container, a transport plane, or a warehouse.

Classic formulation (statement)

Fit the most valuable set of items without exceeding knapsack capacity

Knapsack Problem
Brute Force Approach

Exhaustive search

✎ Consider all possible item subsets

✎ Calculate weight for each

✎ Pick a highest-value subset that fits

Quiz
In the example, how many subsets are there to consider? How do we know in general (for *n* items)?

Example (later)

$W, \{w_i\}, \{v_i\}$

$$W = 10$$
$$w = \{3, 6, 5, 2\}$$
$$v = \{15, 45, 35, 40\}$$

cs681fig9-10.cdr
Tuesday, March 12, 2024 3:26:41 PM
Color profile: Disabled

Composite Default screen

Combinatorial Objects

➪ **Index set**

? ➪ **Power set**

Items from a finite set may be combined in many ways to create interesting objects, which can be expressed as sequences of item indices.

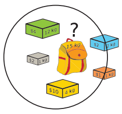
➪ Finite sets

Index set {1,2,3,4,…,*n*}

➪ 2 generation questions

✎ List set members: *permutations*

123 132 312 321 231 213



✎ Group set members: *subsets*

ϕ {1} {2} {3} {1,2} {1,3} {2,3} {1,2,3}

Counting Review

Quiz
Which <u>question</u> answers the quiz about the number of subsets to consider in the example? *Tricky, think about it (answer below)*

3 great combinatorial questions for any set of *n* items

❶ How many ways to order?

Count (1,2) as different than (2,1).

❷ How many pairs (ways to pair)?

Quiz
Why did selection sort, worst-case insertion sort and unique elements have <u>exactly</u> the same number of basic ops?

❸ How many <u>distinct</u> pairs?

*None of them directly,
however, the 3rd helps.
Follow the quiz in the
slide ahead titled „An
Optimal Set".*

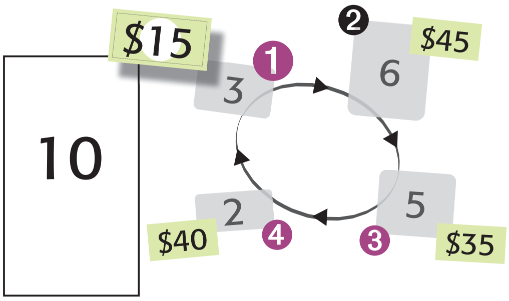
The Knapsack Problem
An Optimal Subset

Exercise
Simulate an algorithm to identify the optimal subset. Hint: go line by line, record latest encountered max (*don't look at the whole table*).

Quiz
(a) Which counting formula is used to count subsets of size two? **(b)** What's the general formula for any size (*i*)? **(c)** Write a sum for <u>total</u> number of subsets.

	Subset	Weight	Value
	ϕ	0	0
	{1}	3	15
	{2}	6	45
	{3}	5	35
	{4}	2	40
i	{1,2}	9	60
	{1,3}	8	50
	{1,4}	5	55
	{2,3}	~~11~~	–
	{2,4}	8	85
	{3,4}	7	75
	{1,2,3}	~~14~~	–
	{1,2,4}	~~11~~	–
	{1,3,4}	10	90
	{2,3,4}	~~13~~	–
	{1,2,3,4}	~~16~~	–

item index set
{1,2,3,4}



➪ **Unique?**

➪ **Efficiency?**

➪ **How bad? *next***

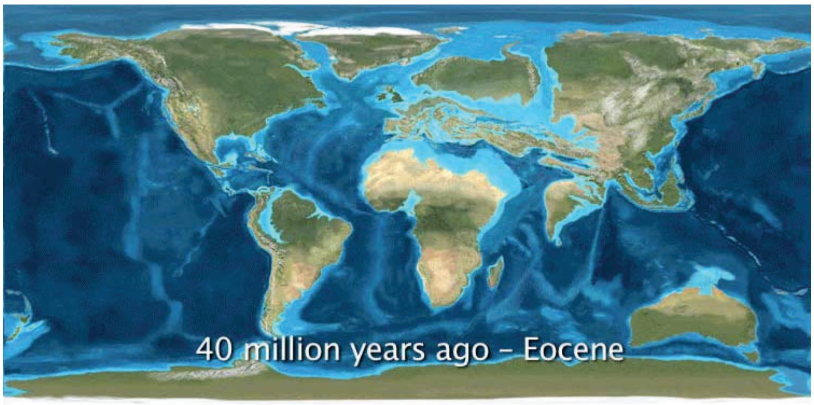
Exponential Growth
A Visualization

Exponential Run Time Visualization			
Input Size	Basic Op Count	Run Time (s)*	Run Time (Yr)
10	1024	<1 0	0
25	33554432	0	0
50	1,125,899,906,842,620	1	0
75	37778931862957200000000	37778932	1
100	1267650600228230000000000000000	1267650600228230	40,170,303

*Assuming a computer that can sustain 1 **petaflops** (10^{15} floating-point operations per second), 1 basic op ≡ average-flop

According to plate tectonics theory, 40 million years ago India had not yet collided (fully perhaps) with Asia!

https://en.wikipedia.org/wiki/Plate_tectonics



Ron Blakey, NAU Geology, CCA-by-SA 4.0 License

Quiz
What if a top (as of 2020) 100 peta-flop supercomputer was used? ***Assume linear scaling, which is not realistic.***

The Traveling Salesman Problem

iStock by Getty Images

	a	b	c	d	e
a	–	3	1	5	8
b	3	–	6	7	9
c	1	6	–	4	2
d	5	7	4	–	3
e	8	9	2	3	–

Levitin, 3rd

Exercise
Specify the formal inputs to the TSP, and give 1–2 instances.

Classic formulation (statement)

Shortest tour through a set of cities visiting each exactly once before returning to the start city

cs681fig9-10.cdr
Tuesday, March 12, 2024 3:26:42 PM
Color profile: Disabled

Composite Default screen

Traveling Salesman Problem
Brute Force Approach

Exhaustive search

- ✎ Consider all possible tours
- ✎ Calculate the length of each tour
- ✎ Pick one with minimum length

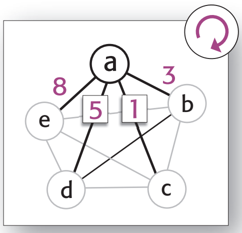
Traveling Salesman Problem

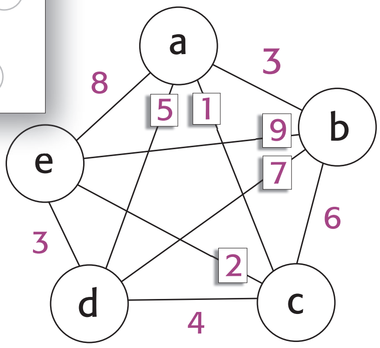
Modeling

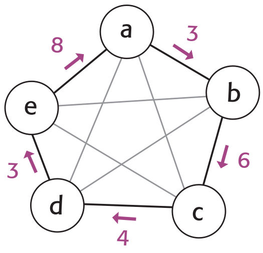
➪ Hamiltonian circuit

Levitin, 3rd

	a	b	c	d	e	
a	–	3	1	5	8	◀
b	3	–	6	7	9	
c	1	6	–	4	2	
d	5	7	4	–	3	
e	8	9	2	3	–	







Quiz
What's the length of tour defined by this HC?

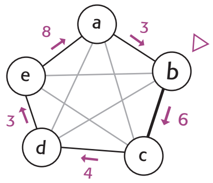
The TSP is converted to a **weighted graph** problem.

Find the best HC

- ✎ Min total weights for shortest tour
- ✎ Largest profit? pick max total weights

Exercise
Lookup other formulations of (ways to state) the TSP. Discuss at least 2. Share in course group.

Traveling Salesman Problem
Hamiltonian Circuits



3 Observations

❶ **Quiz**
Check the tour length of **abcdea** above if it starts at **b** and write the resulting HC. Any different from TSP view?

	1	2	3	4	5
	▽	▽	▽	▽	▽
	a	b	c	d	e
a	–	3	1	5	8
b	3	–	6	7	9
c	1	6	–	4	2
d	5	7	4	–	3
e	8	9	2	3	–

Quiz
What's the combinatorial objects in this case? How many HC start/end at **a**?

Exercise
Write HC starting with **b**, compare to figure. (Try a few, match with ones in figure).

❷
n+1 vertices (n−1 distinct)

1,2,3,4,5,1	24	1,5,2,4,3,1	29
▶ 1,**3**,**2**,4,5,1 ×	25	1,2,5,4,3,1	20
1,4,2,3,5,1	28	1,4,5,2,3,1	24
1,2,4,3,5,1	24	1,5,4,2,3,1	25
1,3,4,2,5,1	29	1,2,4,5,3,1	16
1,4,3,2,5,1	32	1,4,2,5,3,1	24
1,4,3,5,2,1	23	1,3,2,5,4,1	24
1,3,4,5,2,1	20	1,2,3,5,4,1	19
1,5,4,3,2,1	24	1,5,3,2,4,1	28
1,4,5,3,2,1	19	1,3,5,2,4,1	24
1,3,5,4,2,1	16	1,2,5,3,4,1	23
1,5,3,4,2,1	24	1,5,2,3,4,1	32

Exercise
Identify the tour of <u>opposite</u> direction to indicated one. Can we safely exclude it?

❸ rule?

Traveling Salesman Problem

An Optimal Tour

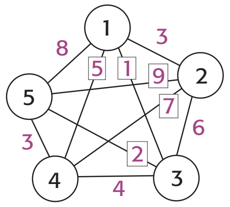
➪ 2-before-3 rule

Essentially, only check <u>distinct tours</u> by eliminating opposite and alternate start-verts versions.

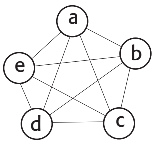
Exercise
Write the component distances of indicated tours.

	1,**2**,**3**,4,5,1	?	
	1,4,2,3,5,1	28	?
	1,2,4,3,5,1	24	
	1,5,2,4,3,1	29	
	1,2,5,4,3,1	20	
	1,4,5,2,3,1	24	
	1,5,4,2,3,1	25	
▶	1,2,4,5,3,1	16	
	1,4,2,5,3,1	24	
	1,2,3,5,4,1	19	?
	1,2,5,3,4,1	23	
	1,5,2,3,4,1	32	

	1 ~~a~~	2 ~~b~~	3 ~~c~~	4 ~~d~~	5 ~~e~~
a	–	3	1	5	8
b	3	–	6	7	9
c	1	6	–	4	2
d	5	7	4	–	3
e	8	9	2	3	–



Exercise
Highlight below the optimal HC, show weights.



➪ Unique?

➪ Count? In general

Traveling Salesman Problem
Efficiency

➪ Exhaustive search: counts

- ✎ All tours: example, general
- ✎ Single start-end city
- ✎ Distinct HC

Start-stop at a city: (n−1)!, therefore all (n−1)! × n = n!, (n−1)!/2 distinct.

Exercise 🕖
How many tours <u>need</u> to be checked if we double cities to 10? How many fold was the increase?

➪ A warning

Example deceptive, only a small number of tours apparently

Another Combinatorial Problem

So far, an optimal solution can be found after checking all solution possibilities which grow combinatorially with input size, i.e., efficiency depends on listing a combinatorial object.

A solution is not readily obvious: cheaper job to assign to Person 2 is Job 3, but cheapest Job 3 is really with Person 3.

The point is a minimum sum not individual costs.

Levitin, 3rd

Cost	Job 1	Job 2	Job 3	Job 4
Person 1	9	2	7	8
▶ Person 2	6	4	▸3	7
Person 3	5	8	1◂	8
Person 4	7	6	9	4

$P_1\ \ P_2\ \ P_3\ \ P_4$

$< 1, 2, 3, 4 >$

Quiz
What's the cost of the trivial (indicated) job assignment?

Assign jobs to persons such that total cost is minimum

The Assignment Problem

Quiz
Identify the combinatorial objects in an exhaustive search solution like KP and TSP? Hint: write another job assignment like in previous slide.

Classic formulation (statement)

✎ Persons (agents), n

✎ Jobs (tasks), n

Possible objectives: minimize financial cost or completion time, maximize satisfaction.

✎ Cost (weight) function + a constraint

Applications

Minimize trip distance or time, fuel cost or consumption.

✎ Assign aircrafts to trips

Maximize utilization, profit, or productivity, or minimize operating cost.

✎ Equipment to facility, salesman to region…

A Closer Look

➪ Optimization problem

Algorithms based on an exhaustive search for such problems tend to be ridiculously simple, in principle.

➪ **Results summary**

Quiz
Suggest an exhaustive search solution to sorting if viewed as a combinatorial problem. Compare to selection sort.

➪ **How bad? Compare sorting**

Best known efficiency?

Exercise
Why is the Assignment Problem fundamentally different even though it has a similar structure and can be solved similarly?

➪ **Assignment vs. the other two**

Known much better efficiency; *none* for KP & TSP!

Why some optimization problems pose difficulty?

➪ **How come?** ***Next*** 🤔

Review

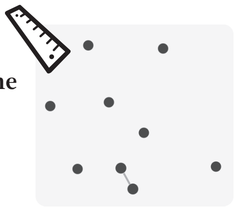
➪ Problem reduction

Why did selection sort, worst-case insertion sort and unique elements have exactly the same number of basic ops?

?

➪ Revisit counting questions (quiz)

Quiz
Describe a **brute-force** approach for finding the closest pair of points.



- ✎ Expected efficiency (what's in common?)
- ✎ Conclusion

Measure distance between all unique (why?) pairs in set of points, i.e., counting Q3 (nC2 = ½(n−1)×n). All algorithms that examine (or enumerate, i.e., a subset or all) distinct pairs of items from a finite set will be at least O(n²).

A **reduction** involves a function that maps (transforms) an instance of *P* to an instance in *Q* for all instances (i.e., get same result from either).

Id original problem *P* and its question, **reduced problem** *Q* and its equivalent question.

Turning a geometric question to an algebraic one is very common.

➪ Transforming the TSP to solve

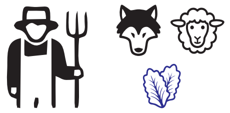
- ✎ Major theoretical importance
- ✎ A thinking map *TSP as example* $P \rightarrow Q$
- ✎ Reduction patterns in algorithms

cs681fig9-10.cdr
Tuesday, March 12, 2024 3:26:49 PM
Color profile: Disabled

Composite Default screen

Problem Reduction
Solve A Famous Puzzle

➪ State-space graph

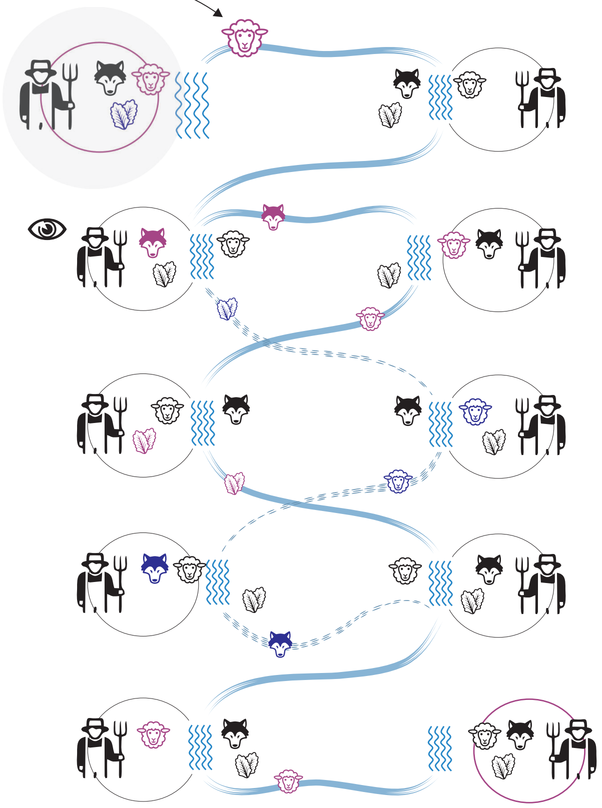


Legal moves leading to valid positions suggest edges and verts, respectively, in a graph (should it be directed?).

Quiz
What is an equivalent question in the (reduced) graph problem *Q*? Suggest a graph algorithm to answer it?

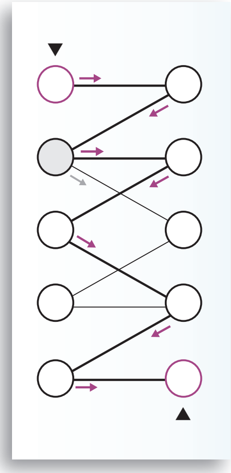
Some problems involve exploring sequences of states representing possible solutions or positions, i.e., searching a **state space** (what if states increase exponentially with input size?).

Exercise
Give examples for reductions to graph problems. **Hint**: not a graph problem but 1st step to solve it is to draw a graph.



P Unknown problem
Q Known problem

$P \rightarrow Q$



Exercise

Pick an activity trail to pursue & report back

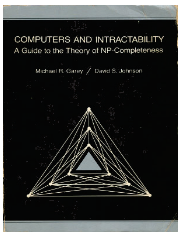
Exhaustive search KP & TSP

- ✎ Programming challenges
- ✎ Optimizations (opts), impact?
- ✎ Specify algorithms (write a pseudocode)

Investigate procedures to generate the required combinatorial objects.

For example, in the KP, perhaps check first if all fit in *O*(*n*), just in case.

Exercise
Check at least 2 opts for KP, discuss effects on efficiency.


COMPUTERS AND INTRACTABILITY
A Guide to the Theory of NP-Completeness
Michael R. Garey / David S. Johnson

Read pp. 1-15 Garey & Johnson

Computers and Intractability: A Guide to the Theory of NP-Completeness