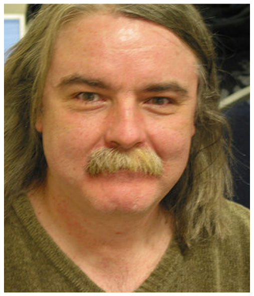
Why is sorting data important?

- Guess a number in [1,100] --- I'll tell you yes or no
  - How many guesses needed worst-case? Average?
  -
- Guess a number, I'll tell you high, low, correct
  - How many guessed needed?
  -
- How do we find the 100 people most like us on …
  - Amazon, Netflix, Facebook, …
  - Need a metric for analyzing closeness/nearness,
  - Need to find people "close", sorting can help

How do we sort? Ask Tim Peters!



- Sorting API
  - Sort lists (or arrays, …)
  - Backwards, forwards, …
  - Change comparison
    - First, Last, combo, …
- Sorting Algorithms
  - We'll use what's standard!

Best quote: import this

I've even been known to get Marmite *near* my mouth -- but never actually in it yet. Vegamite is right out

APTs Sorted and Sortby Frequencies

- What are the organizing principles in SortedFreqs?
  - Alphabetized list of unique words?
  - Count of number of times each occurs?
  - Is efficiency an issue? If so what recourse?
  - To create a list comprehension:
    - # elements in resulting list? Type of elements in list?
- What are organizing principles in SortByFreqs?
  - How do we sort by frequency?
  - How do we break ties?

```
sorted([(t[1],t[0]) for t in dict.items()])
sorted(dict.items(), key=operator.itemgetter(1))
```

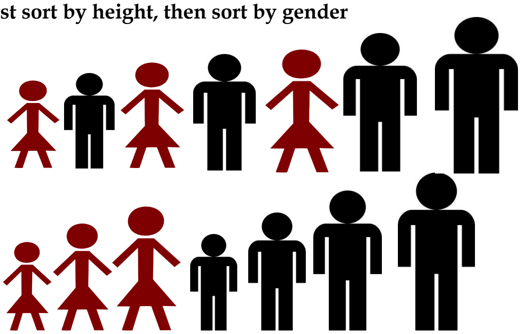
Sorting from an API/Client perspective

- API is Application Programming Interface, what is this for sorted(..) and .sort() in Python?
  - Sorting algorithm is efficient, part of API?
  - Sorting algorithm is stable, part of API?
  - `sorted(list,reverse=True)`, part of API
- Idiom:
  - Sort by two criteria: use a two-pass sort, first is secondary criteria (e.g., break ties)

```
[("ant",5),("bat", 4),("cat",5),("dog",4)]
[("ant",5),("cat", 5),("bat",4),("dog",4)]
```

Stable sorting: respect re-order

- Women before men …
  - First sort by height, then sort by gender



Python Sort API by example, (see APT)

- Sort by frequency, break ties alphabetically

```
def sort(data):
  d = {}
  for w in data:
    d[w] = d.get(w,0) + 1
  ts = sorted([(p[1],p[0]) for p in d.items()])
  #print ts
  return [t[1] for t in ts]
```

- How to change to high-to-low: `reverse=True`
- How to do two-pass: `itemgetter(1)` from operator

How to import

- We can write: import operator
  - Then use key=operator.itemgetter(…)
- We can write: from operator import itemgetter
  - Then use key=itemgetter(…)
- From math import pow, From cannon import pow
  - Oops, better not to do that
- Why is import this funny?
  - This has a special meaning

Organization facilitates search

- How does Google find matching web pages?
  -
- How does Soundhound find your song?
  -
- How does tineye find your image
  -
- How do you search in a "real" dictionary
  -
- How do you search a list of sorted stuff?
  - `bisect.bisect_left(list,elt)`