DMSS | 13 | 21-12-16

(comma) Str. match p.985

Today

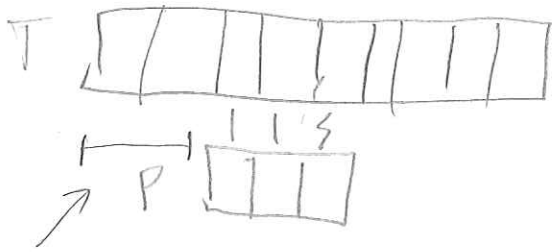
Last time

Next will be "Spectre"
(will make doodle)

String matching

String matching

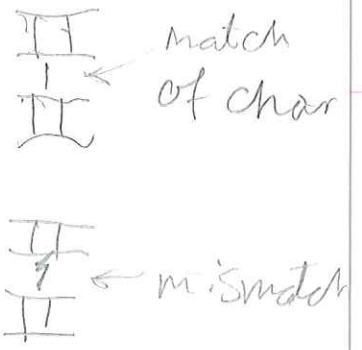
Σ : alphabet



Sliding window, shift

$|T| = n$;

$|P| = m$;



Naive:

Shift with 1 each time

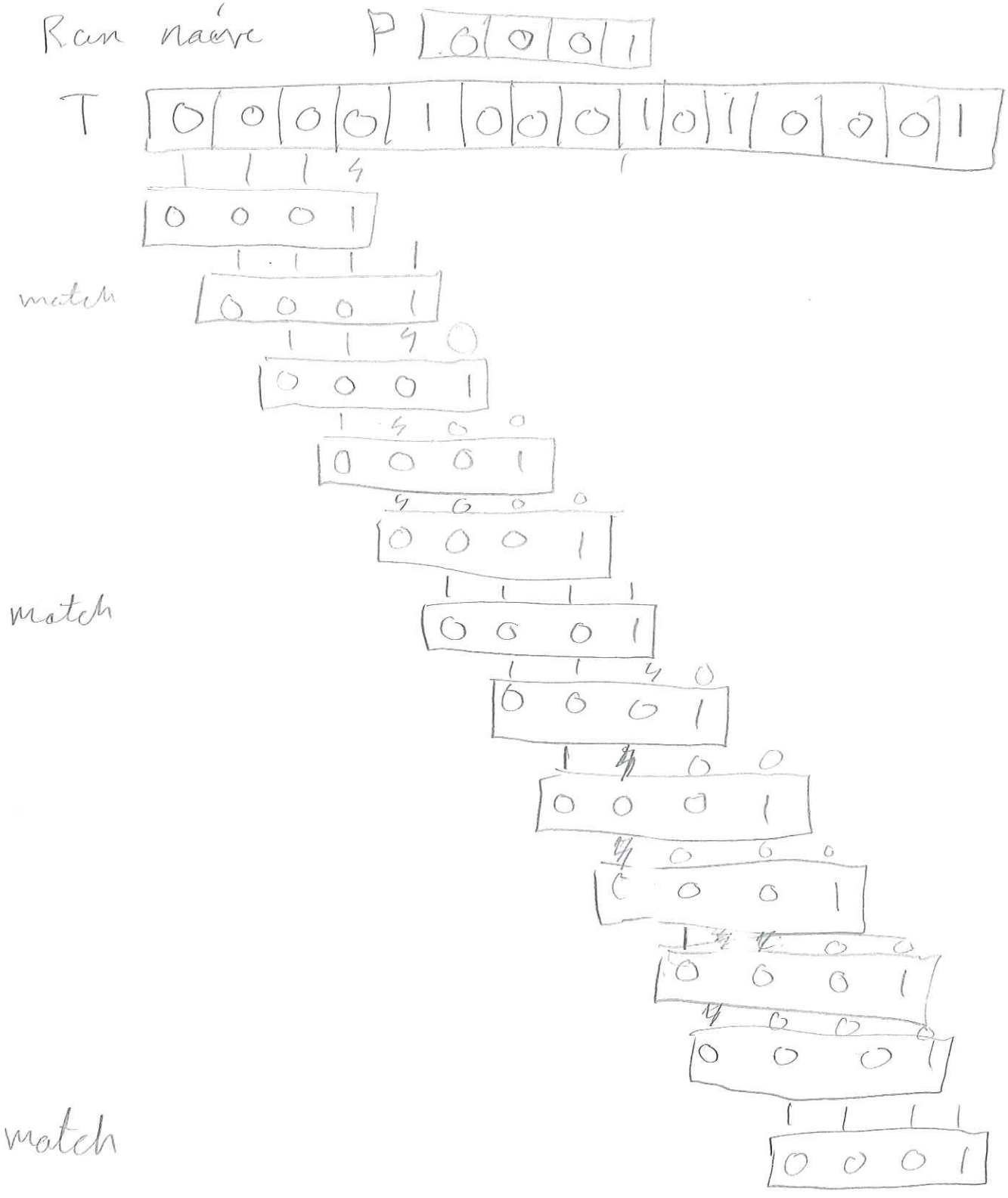
can take $O(n \cdot m)$

if T a|a|a|a|a...

a|a|a

Comment 32.1-1

Run naïve



Cormen 32.1-4

P can now have <u>gap char</u> ◇

(In DMSS3 known as Σ^*)

For instance P | a | b | ◇ | b | a |

will match T | a | c | a | b | c | c | b | a |

This part could also just be here

| a | b | b | a |

| a | b | a | b | a |

| a | b | a | a | b | a |

| a | b | b | a | b | a | b | a |

And ◇ can appear an arbitrary # of times in P

▷ Give poly time alg to do this (only 1 occurrence necessary)

one idea:

~~Find~~ Split P up into $P_1 \ldots P_k$ on ◇

so | a | b | ◇ | b | a | ◇ | c |

P_1 P_2 P_3

Find P_1 in T; then find P_2 somewhere in the rest of T, ..., P_k in last part of T

$O(n \cdot m \cdot k)$, definitely upper bound

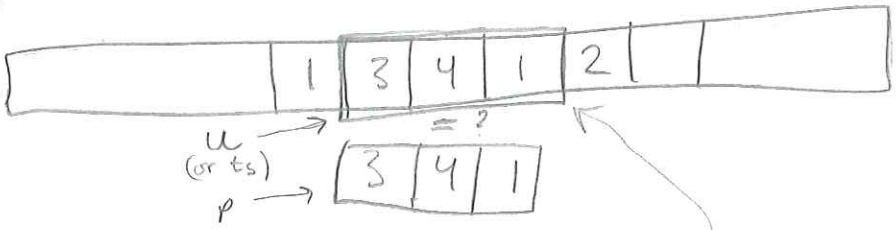
$O(n \cdot m)$ might work, if the combined work on all P_i is just original P (minus the ◇)

Cormen 32.2-2

Rabin-Karp video: bit.ly/2hErrjg

$\Sigma = \{0, 1, \ldots, 9\}$

Consider each sliding window a big base 10 number and compare with pattern also as base 10 number



If we can get slice of T as base 10 number in $O(1)$ and compare 2 base 10 in $O(1)$ then this gives $O(n+m)$.

▷ Slice of T

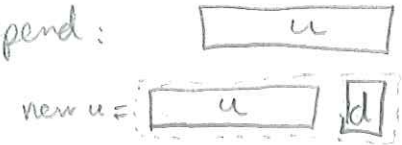
Suppose we had [3|4|1] as number 341 (u)

How could we remove [3] and add [2] ? (the next in T)

Horners rule!

2 steps

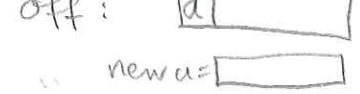
① Append:



$$\text{new } u = \underbrace{u \cdot 10}_{\text{shift}} + d$$

$341 \cdot 10 + 2$
$= 3410 + 2$
$= 3412$

② Pop off:



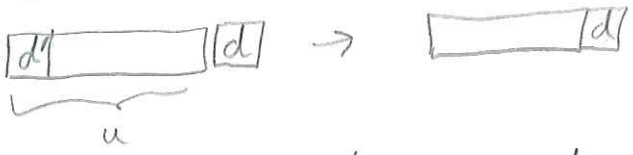
$$\text{new } u = u - d' \cdot 10^{m-1}$$

$3412 - 3 \cdot 10^{4-1}$
$= 3412 - 3000$
$= 412$

Continued

Cormen 32.2-2 continued

Horners rule (combined):



$$\text{new } u = (u - d' \cdot \underbrace{10^{m-1}}_{\text{pop}}) \cdot 10 + d$$

▷ Now we can get to next window in $O(1)$ if we precompute 10^{m-1}.
We call this value a rolling value.

▷ Compare in $O(1)$:
u and p can be as big as m which would ruin the time

Solution: work modulo prime q ← such that $10q \leq \underbrace{2^{64}}_{\text{computer word}}$

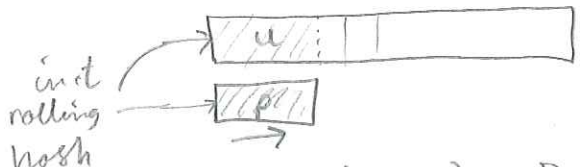
rolling hash

$$\text{new } u = (u - d' \cdot h) \cdot 10 + d \mod q$$

$h = 10^{m-1} \mod q$ (precompute)

▷ Now we can move to next and check for equality in $O(1)$. But multiple things can hash to the same thing!

Alg in short:



for both (p never change). Do naive-like shifts using Horners rule.
If match, do char-by-char to make sure it is correct. This costs $O(m)$, but if there is match it is okay. And no-match when $u = p$ only happen with prob $\frac{1}{q}$, so exp no-match cost is $\frac{nm}{q}$.
Alternatively one can say this happens $\frac{n}{q}$ times of cost $O(m)$

Cormen 32.2-2

▷ Now for the exercise:

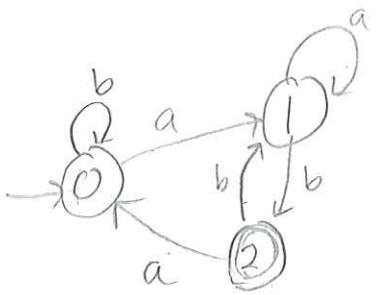
Extend Rabin-Karp to search for multiple (k) patterns at once

▷ we have hash of each
what if we store them in hash table T?

▷ Now for any u we check if $u \in T$
If it is check T slice against all P's
to find if any match
This runs in $O_E(n + km)$ which is better than $O((n+m) \cdot k)$ naive

Cormen 32.3 - 2

DFA



Digraph of states
Following edges consumes char from input
(A state must have an out-edge for each char $\in \Sigma$)

→○: start state

◎: report match

Above DFA matches, for instance:

a b, b a a b, b b b a b b a b, b a b a a b

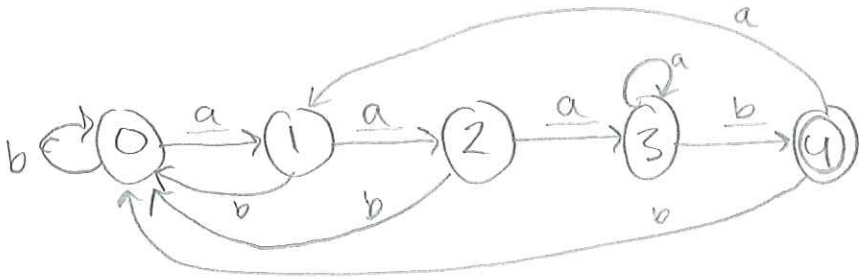
String matching DFA

Preprocess: create DFA from P

Run DFA on T

The general idea of how DFA looks

P | a | a | a | b | becomes



If everything just matches it follows a a a b.
When things mismatch we reuse as much as possible of what we have already matched.

Alg:

```
m = P.length
for q = 0 to m
    for c ∈ Σ
        k = min(m+1, q+2)
        repeat
            k = k-1
        until P[1..k] ⊐ P[1..q] + c
        δ(q,c) = k
return δ
```

⊐: is suffix of
+: str concat

Cormen 32.3-2

This is shortened ↓ (Actually 21 char)

▷ Exercise: Construct DFA for P = a b a b b a b (m = 7)

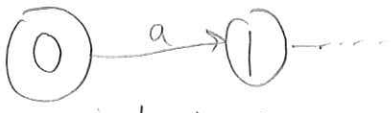
▷ How many states?

↳ 8, 0–7

(0) (1) (2) (3) (4) (5) (6) (7)

we can fill in match from intuition

Now for the rest.

▷ 

what about b?

q = 0
c = b

k = 0 + 2 = 2
k = k − 1 = 2 − 1 = 1

[a] ? [b] %

P[1..k] P[1..q] c

k = k − 1 = 1 − 1 = 0

[] ? [b] ✓ (This is always true, empty str is suffix of any str)

δ(0, b) = k = 0

So

▷ 

what about a?

q = 1 ; k = 3 ; P[1..q] + c = [a a] (doesnt change)

c = a ; k = 2

[a b] ? [a a] %

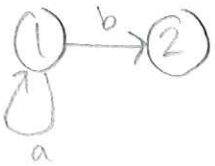
P[1..k]

k = 1

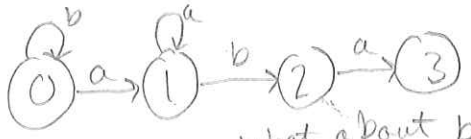
[a] ? [a a] ✓

δ(1, a) = k = 1

So



Cormen 32.3-2

▷ 

$q = 2$ $K = 4$ $P[1..q]+c =$

$c = b$ $k = 3$

| a | b | b |

| a | b | a |

Can we see with intuition how far down we need before ∃-check is true?

| a | b | a | ✗

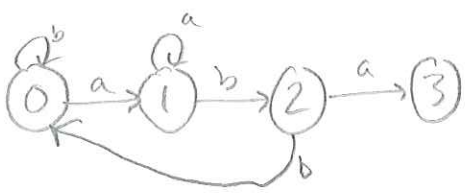
| a | b | ✗

| a | ✗

∄ ✓ so $k = 0$

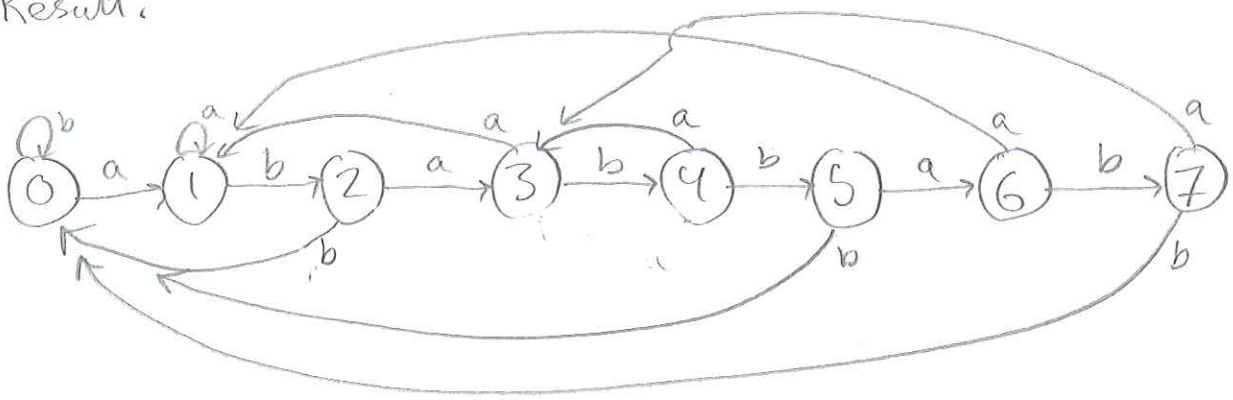
$\delta(2, b) = 0$

So



▷ ... And so on

▷ Result:



Cormen 32.3-4

2 patterns P, P'

construct DFA to find all occurrences of either pattern

▷ (we can get DFA for both of them separately)

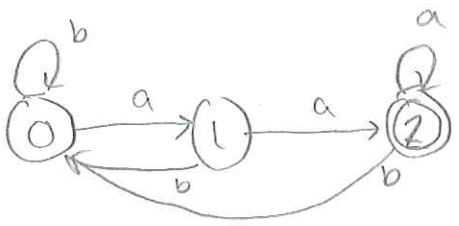
▷ In DM553 you will see two ways to do this:
Cross product and by NFA with ε moves.

We do cross product here by example.

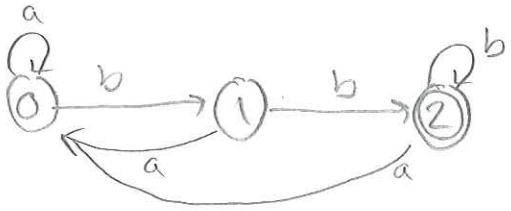
Idea: Simulate both DFAs at the same time in one DFA.
States will be (q, q'), where q is state from DFA of P and q' is state DFA of P'.

Example:

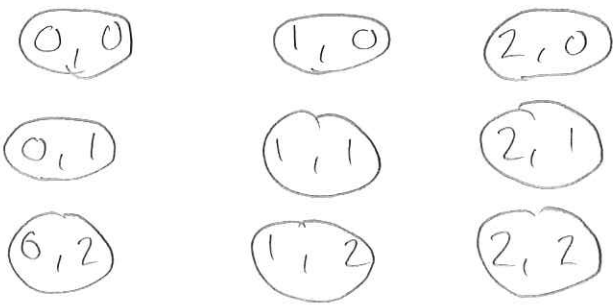
P [a | a] $\Rightarrow$



P' [b | b] $\Rightarrow$

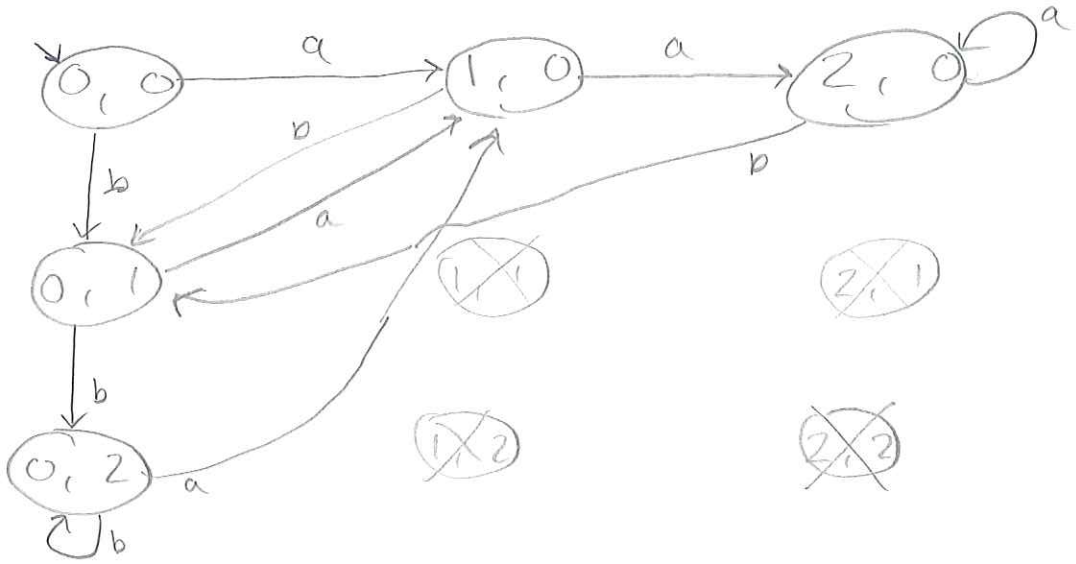


States:



Cormen 32.3-4

▷ Putting in transitions



▷ Accept states?

(2,0) and (0, 2)

Jan 10.3 Multiple choice

2 choices per Question. 10 questions

conditions:

- All Qs must be answered
- wrong answers count towards you.

  n_1 is right As, n_2 is wrong As

  $n_1 - n_2$ is score.

  For 10 Qs and n_1 right, score is $n_1 - (10 - n_1)$

  $= 2n_1 - 10$

▷ 3 students

A: $\frac{3}{5}$ for correct, X_A = Score of A

B: $\frac{4}{5}$ for correct, X_B = Score of B

C: $\frac{1}{2}$ for correct, X_C = Score of C

a) $E(X_A)$? we expect $10 \cdot \frac{3}{5}$ to be correct and $10 \cdot \frac{2}{5}$ to be wrong.

$\Rightarrow 10 \cdot \frac{3}{5} - 10\frac{2}{5} = 2$

$E(X_B) = 10 \cdot \frac{4}{5} - 10 \cdot \frac{1}{5} = 6$

$E(X_C) = 10\frac{1}{2} - 10 \cdot \frac{1}{2} = 0$

b) $P(X_C > E(X_C))$? I.e. $P(X_C > 0)$? As it is completely symmetrical around 0, $P(X_C > 0) = P(X_C < 0)$, so $P(X_C \neq 0) = 2 \cdot P(X_C > 0)$

$\Rightarrow P(X_C = 0) = 1 - 2 \cdot P(X_C > 0) \Rightarrow P(X_C > 0) = \frac{1 - P(X_C = 0)}{2}$

Now we focus on $P(X_C = 0)$, ie. prob that $n_1 = n_2$

C outputs a random bit str,

And all $\binom{10}{5}$ ways to choose bitstr with 5 ones (and 5 zeros)

$\Rightarrow$ Prob $= \frac{\binom{10}{5}}{2^{10}} = \frac{63}{256}$; Result: $\frac{1 - \frac{63}{256}}{2} = \frac{193}{512}$. continued

Jan 10.3 continued

c) Bar for passing is now 2 points
$P(A \text{ pass})$? $P(B \text{ pass})$? $P(C \text{ pass})$?

Note: Getting 2 points means $n_1 = 6$, $n_2 = 4$
Also: Not possible to get 1 point.

We already know $P(C \text{ pass})$ as it is $P(X_c > 0) = P(X_c \geq 2) = \frac{193}{512}$.

We can use Bernoulli for the rest

[Prob of exactly k success in n trial, where prob of success is p and failure is q (p+q=1) is $\binom{n}{k} p^k q^{n-k}$

We want 6, 7, 8, 9 or 10 success/rights:

A: $$\sum_{k=6}^{10} \binom{16}{k} \left(\frac{3}{5}\right)^k \left(\frac{2}{5}\right)^{16-k} = \frac{6182649}{9765625} \approx 0,633$$

B: $$\sum_{k=6}^{10} \binom{10}{k} \left(\frac{4}{5}\right)^k \left(\frac{1}{5}\right)^{10-k} = \frac{9445376}{9765625} \approx 0,967$$

d) We now have 8 students and 6 that are guessing
What is the expected # that pass?
Each is just indep trial ⇒ use exp Bernoulli
Chance of success $p = \frac{193}{512}$, Result: $8 \cdot \frac{193}{512} \approx 3,016$

e) Recap: Chernoff bounds
If $X = X_1 + X_2 + \cdots + X_n$, and all X_i are indep
we can bound:

$$P(X > (1+\delta)\mu) < \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^\mu$$

where $\mu \geq E(X)$ (think of it as just $E(X)$)
and $\delta > 0$ is just some chosen number.

continued

Jan 10.3 continued

e) we now consider 3 different tests using same system:

- 100 Qs
- 200 Qs
- 500 Qs

Pass: 20% score

We now want to bound the prob that C passes each.

Let $Y_i = \begin{cases} 1 & \text{if C answers correctly on Q}i \\ 0 & \text{o.w.} \end{cases}$ / $Y = \sum_{i=1}^{n} Y_i$

$E(Y_i) = P(Y_i = 1) = \frac{1}{2}$; $E(Y) = \frac{1}{2} n$

We can now use Chernoff as Y_i indep

$\delta = 0,2 \qquad \mu = \frac{1}{2} n$

$$P(Y > \underbrace{(1+0,2)\tfrac{1}{2} n}_{\text{20\% more than expected (0 points)}}) < \left(\frac{e^{0,2}}{(1+0,2)^{(1+0,2)}} \right)^{\frac{1}{2} n} \approx 0,9814^{\frac{1}{2} n}$$

$n = 100$: $\left(\frac{e^{0,2}}{1,2^{1,2}} \right)^{50} \approx 0,3909$

$n = 200$: $\left(\frac{e^{0,2}}{1,2^{1,2}} \right)^{100} \approx 0,1528$

$n = 500$: $\left(\frac{e^{0,2}}{1,2^{1,2}} \right)^{250} \approx 0,0091$

As expected, the chance of passing decreases as the number of Qs increase. It gets harder to get away from $E(Y)$!

Answer 13.4

Similar to one of the Cormen exercises about universal hashing.

uniform: $\forall x \in U$, $\forall z \in \mathbb{Z}_p$ and h random from H

$$P(h(x) = z) = \frac{1}{p}$$

consider H:

$$h_a(x) = \left(\sum_{i=1}^{n} a_i x_i \right) \bmod p$$

| key x chopped up in blocks x_i

we have seen that it is universal, but is it uniform? (Answer: no) but why?

▷ Can we find key x that map to some z no matter choice of h_a?

↳ $x = 0$, it will always map to 0.
So it is not entirely uniform $\frac{1}{p}$