Another approach

Dan Gusfield's *Z* values

The Z values of a string S

- $Z(i)$ of a string S is the largest integer d such that $S[1\ldots d] = S[i\ldots i+d-1]$

Clearly, …

- If $Z(1)$, $Z(2)$, …, $Z(|S|)$ are the Z values of S, then
  - $Z(1) = |S|$;
  - $Z(i) \geq 0$ for each $i = 1, 2, \ldots, |S|$.

For example, …

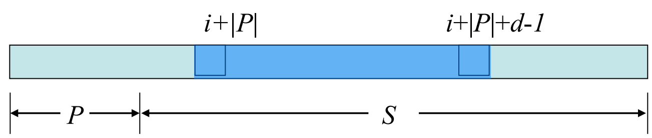
```
S =  a a g c a a t a a a g c
Z = 12 1 0 0 2 1 0 2 4 1 0 0
       a     a a   a a
               a     a a g c
                       a
```

Question

How do we find all occurrences of *P* in *S* using *Z* values?

Exact string matching with Z values

```
computing Z values of PS;
for i=1 to |S|
  if Z(i+|P|)≥ |P| then
    output i;
```



Time complexity?

```
computing Z values of PS;
for i=1 to |S|
  if Z(i+n)>=|P| then
    output i;
```

- $O(|S|)$ + time for computing the Z values of PS.

Z(i) can be naively computed in $\Omega(|S|)$ time

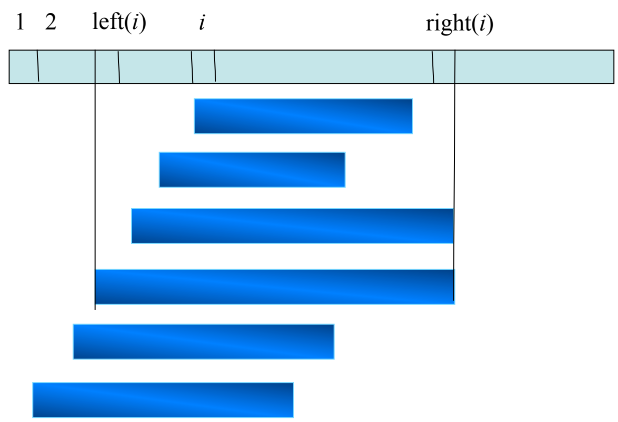
```
For i=1 to |S| {
  set j = i;
  set Z(i) = 0;
  while (S[j]==S[j-i+1]){
    Z(i)++;
    j++;
  }
}
```

Calculate Z values in linear time

Notation

- right(i) = $\max\{j + Z(j) - 1 \mid 1 < j \leq i\}$.
- left(i) = $\min\{j \mid \text{right}(j) = \text{right}(i)\}$.
- Observation: left(i) and right(i) nondecreasing.

Illustration



Strategy

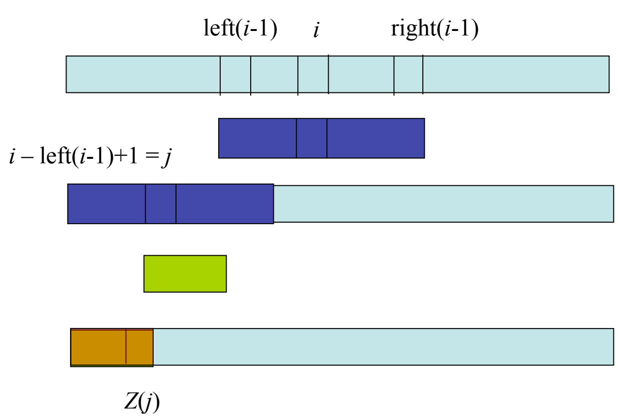
- Computing $Z(i)$, right(i), left(i) from
  - $Z(1)$, $Z(2)$, …, $Z(i – 1)$;
  - right($i – 1$);
  - left($i – 1$).

Case 1: right(i-1) ≤ i-1.

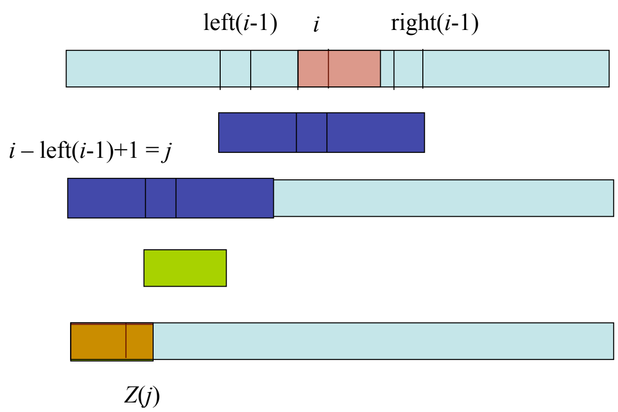
- right(i-1) does not cover i.
- Computing $Z(i)$ naively in $O(1+Z(i))$ time.
- left(i) = i.
- right(i) = $i + Z(i) - 1$.
- Observation

  1+$Z(i)$ = right(i) – i + 2 ≤ right(i) – right(i – 1) + 1.

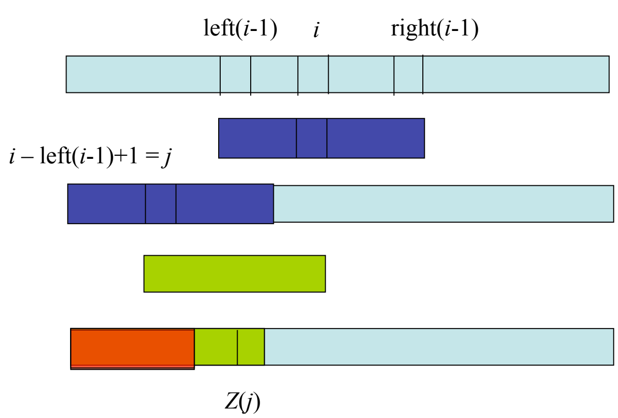
Case 2:right(i-1) ≥ i and Z(j) < right(i-1)-i+1.



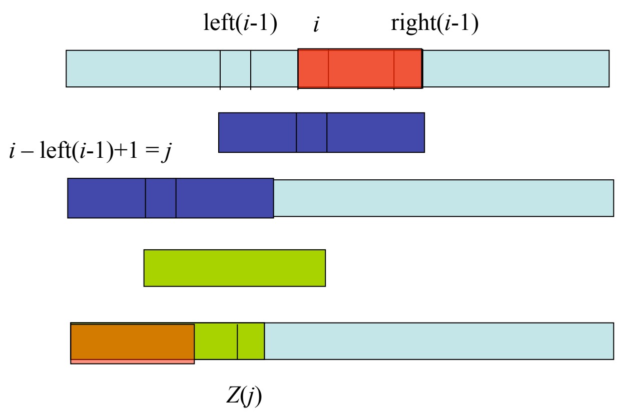
Z(i) = Z(j), left(i) = left(i-1), right(i) = right(i-1).



Case 3:right(i-1) ≥ i and Z(j) ≥ right(i-1)-i+1.

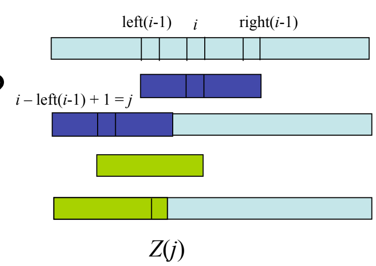


Finding Z(i) by comparisons starting from right(i-1)+1.



Computing left(i) and right(i).

- right(i) = i + Z(i) -1.
- left(i) = i.
- How many comparisons?
  - right(i)-right(i-1)+1.



Time complexity is linear

- Case 1:
  - $O(|Z(i)|+1) = O(\text{right}(i)-\text{right}(i-1)+1)$.
- Case 2:
  - $O(1) = O(\text{right}(i)-\text{right}(i-1)+1)$.
- Case 3:
  - $O(\text{right}(i)-\text{right}(i-1)+1)$.

Overall time complexity: $O(|S|)$

Z-value Pseudo code

```
int L = 0, R = 0;
for (int i = 1; i < n; i++) {
  if (i > R) {
    L = R = i;
    while (R < n && s[R-L] == s[R]) R++;
    z[i] = R-L; R--;
  } else {
    int k = i-L;
    if (z[k] < R-i+1) z[i] = z[k];
    else {
      L = i;
      while (R < n && s[R-L] == s[R]) R++;
      z[i] = R-L; R--;
    }
  }
}
```

Dr. Gusfield’s lecture

- https://www.youtube.com/watch?v=MFK0WYeVEag