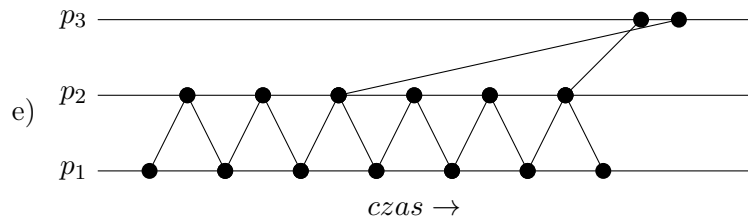
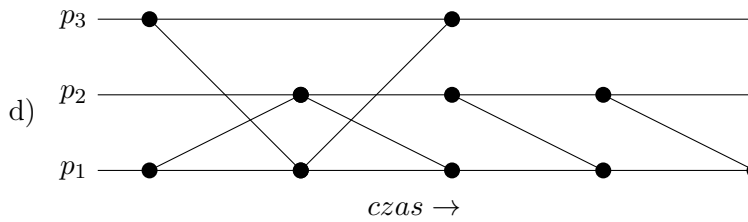
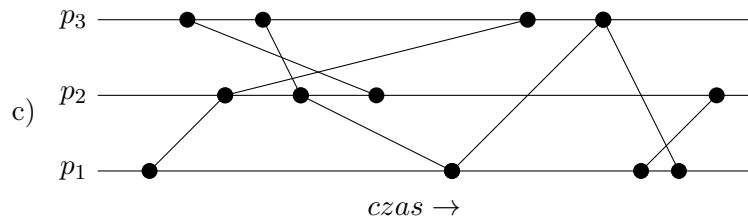
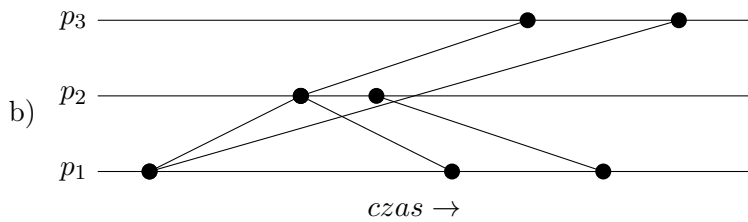
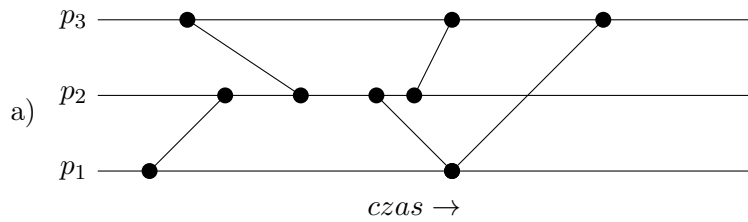


Algorytmy rozproszone

Lista 2

1. (5pt) Rozważmy diagramy wykonania (execution) w modelu przesyłania wiadomości dla trzech procesów p_1 , p_2 , p_3 . Na każdym diagramie opisz czasy zgodnie z definicją zegara logicznego wykorzystując raz zegar Lamport'a i raz zegar wektorowy.



2. (10pt) Rozważmy model przesyłania wiadomości (message-passing) na dwukierunkowym pierścieniu z $n = mk$ węzłami, gdzie $m > 1$ oraz k jest nieparzyste. Węzły na pozycjach $0, k, 2k, \dots, (m-1)k$ są specjalne i nazywamy je „korzenie” (root). Wszystkie węzły mogą odróżnić lewego sąsiada od prawego sąsiada, ale nie znają ich identyfikatorów. Rozważmy Algorytm 1 do rekrutacji węzłów, czyli zaczynając od korzeni tworzymy drzewa łączące sąsiadów.

Algorithm 1: Algorytm rekrutacji

```

1 initially do
2   if jestem korzeniem then
3      $parent \leftarrow id$ 
4     send  $M$  do swoich dwóch sąsiadów
5   else
6      $parent \leftarrow \perp$ 
7 upon receiving  $M$  od sąsiada  $p$  do
8   if  $parent = \perp$  then
9      $parent \leftarrow p$ 
10    send  $M$  do sąsiada (który nie jest  $p$ )

```

- Założmy, że uruchamiamy ten algorytm w systemie synchronicznym. Jaka jest minimalna i maksymalna wielkość drzewa (wielkość, czyli liczba węzłów)?
- Założmy, teraz że algorytm uruchamiamy w systemie asynchronicznym. Jaka jest minimalna i maksymalna wielkość drzewa?
- Podaj algorytm w modelu asynchronicznym, który gwarantuje że wszystkie powstałe drzewa są tej samej wielkości

Do każdego z powyższych podpunktów przeprowadź krótki dowód.

3. (15pt) Mając graf $G = (V, E)$ oznaczmy przez $\epsilon(v)$ maksymalną odległość wierzchołka v od każdego innego wierzchołka w grafie, czyli $\epsilon(v) = \max_{v'} d(v, v')$. Założmy, że mamy system przesyłania wiadomości oraz graf sieci ma strukturę drzewa.
- Zaprojektuj algorytm który pozwoli każdemu węzłowi v obliczyć $\epsilon(v)$.
 - Udowodnij wykorzystując niezmiennik, że wartości obliczone przez zaproponowany algorytm faktycznie poprawnie liczą $\epsilon(v)$ dla każdego v .