Ε.Μ.Π., Σχολή Η.Μ.& Μ.Υ.

Ημερ/νια: 12-04-2021

Συστήματα Αναμονής (Queuing Systems) - Ακαδ.Ετος 2020-2021

<u>1η Ομάδα Ασκήσεων</u>

1. Θεωρητική μελέτη της ουράς M/M/1
2. Ανάλυση ουράς M/M/1 με Octave
3. Διαδικασία γεννήσεων θανάτων (birth-death process): εφαρμογή σε σύστημα M/M/1/K

Παραδοτέα: 26-04-2021

Του προπτυχιακού φοιτητή:

Νικήτας Τσίννας, ΑΜ : 03118187

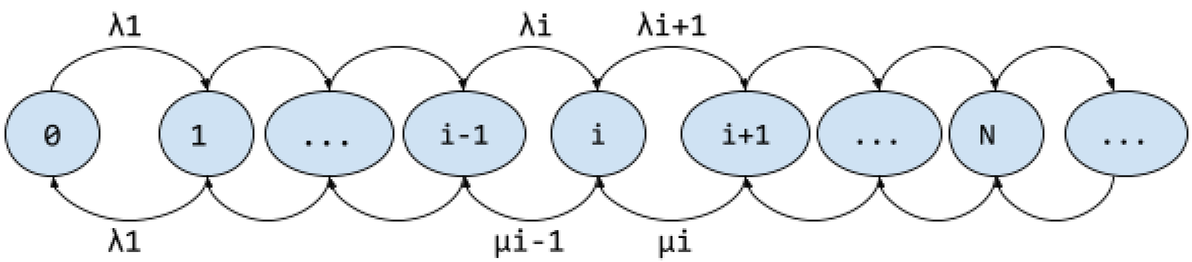
<u>Θεωρητική μελέτη της ουράς M/M/1</u>

α) Γνωρίζουμε πως η απαραίτητη συνθήκη ώστε η ουρά M/M/1 να είναι εργοδική είναι ο ρυθμός των εξυπηρετήσεων να είναι μεγαλύτερος του ρυθμού αφίξεων στην εξυπηρετητή. Δηλαδή πρέπει να ισχύει:

$$\boldsymbol{\rho = \lambda/\mu < 1}$$

Όπου λ η παράμετρος της κατανομής Poisson και μ η παράμετρος της εκθετικής κατανομής.

Ακολουθεί το διάγραμμα ρυθμού μεταβάσεων της M/M/1:



Στην περίπτωσή μας, ισχύει πως σε όλες τις καταστάσεις η παράμετρος Poisson των αφίξεων είναι ίδια, καθώς και η παράμετρος της εκθετικής κατανομής των εξυπηρετήσεων. Άρα:

- $\lambda_1 = \ldots = \lambda_i = \lambda_{i+1} = \ldots = \lambda_N = \boldsymbol{\lambda}$
- $\mu_0 = \ldots = \mu_{i-1} = \mu_i = \ldots = \mu_{N-1} = \boldsymbol{\mu}$

Αν χρησιμοποιήσουμε την σφαιρική και τοπική εξίσωση ισορροπίας
- $\lambda P_{i-1} = \mu P_i$, $i = 1, 2, 3, \ldots$
- $(\lambda_k + \mu_k)P_k = \lambda_{k-1}P_{k-1} + \mu_{k+1}P_{k+1}$, $k = 1, 2, 3, \ldots$

Τότε, $\lambda * P0 = \mu * P1 \Rightarrow P1 = (\lambda/\mu) * P0 = \rho * P0$

και $(\lambda + \mu)P1 = \lambda P0 + \mu P2 \Rightarrow P2 = \rho^2 P0 \Rightarrow Pk = \rho^\kappa P0$

Άρα χρησιμοποιώντας την προηγούμενη στην 3η εξίσωση ισορροπίας:

$P0 + P1 + \ldots + PN + \ldots = P0(\rho + \rho^2 + \ldots + \rho^N + \ldots)$, τότε αν 0<ρ<1 τότε η παραπάνω δυναμοσειρά συγκλίνει και έτσι
P0 * (1/(1-ρ)) = 1 => P0 = 1-ρ και Pk = (1-ρ)*ρ^k

β) Γνωρίζουμε πως μπορούμε να υπολογίσουμε τον μέσο χρόνο καθυστέρησης ενός πελάτη στο σύστημα, όταν η ουρά αναμονής βρίσκεται σε ισορροπία, χρησιμοποιώντας τον νόμο του Little. Δηλαδή

$$E(T) = E[n(t)]/\gamma = E[n(t)]/\lambda],$$

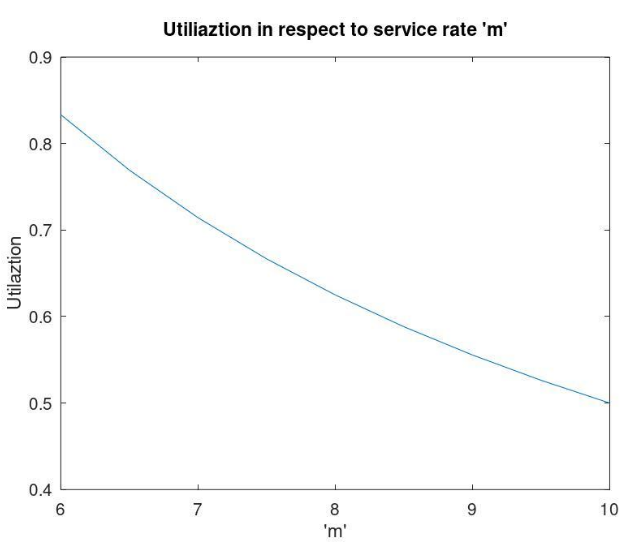
εφόσον η ουρά έχει άπειρη χωρητικότητα. Επομένως,
E(T) = ρ/[(1-ρ)*λ] (ρ=λ/μ) => **E(T) = 1/(μ-λ)**

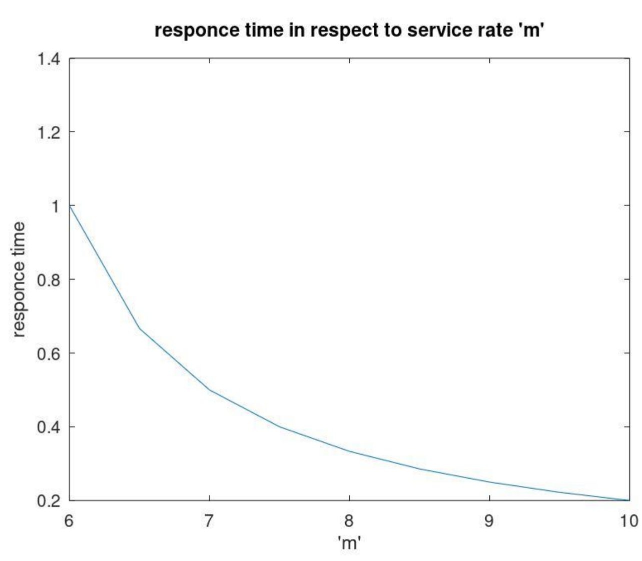
γ) Πρακτικά, ζητείται η πιθανότητα κατάστασης P_{57} . Όπως υπολογίστηκε παραπάνω, θα είναι $P_{57} = (1-\rho)\rho^{57}$. Δηλαδή η πιθανότητα είναι θετική άρα υπαρκτή. Επίσης, σχετίζεται άμεσα με το ρ. Όσο μεγαλύτερος είναι ο ρυθμός εξυπηρετήσεων και όσο μικρότερος είναι ο ρυθμός αφίξεων, τόσο μικρότερη θα είναι η πιθανότητα να βρεθεί το σύστημα σε μία κατάσταση P57 και για κάθε κατάσταση με μεγάλο k.

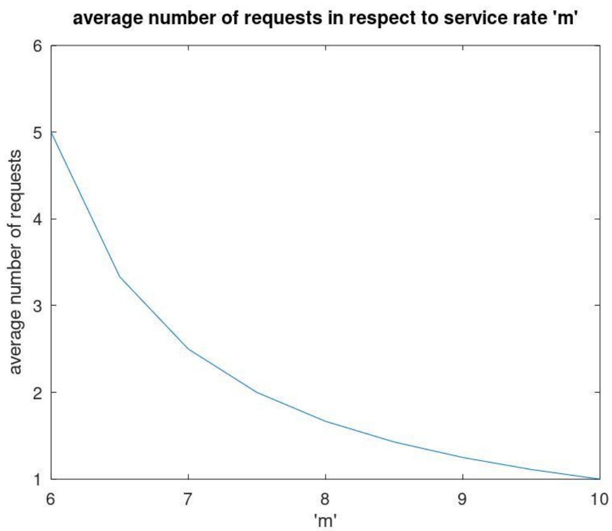
<u>Ανάλυση ουράς M/M/1 με Octave</u>

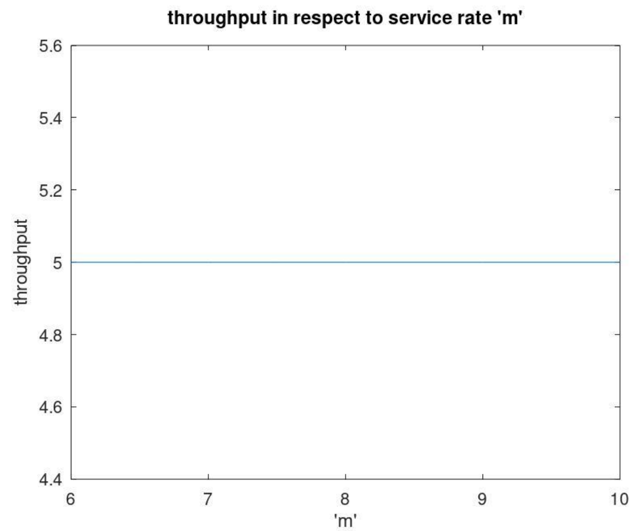
α) Σύμφωνα με το πρώτο ερώτημα της πρώτης άσκησης, θα πρέπει ρ<1, δηλαδή λ/μ<1 ή μ>λ. Άρα θα πρέπει μ>5. Άρα μπορούμε να επιλέξουμε εξυπηρετητή με μ=[6,10] πελάτες/min.

β) Ακολουθούν τα ζητούμενα διαγράμματα:









γ) Παρατηρώ στο διάγραμμα του μέσου χρόνου καθυστέρησης (response time) πως στο διάστημα για μ=[7,9] ο χρόνος καθυστέρησης ομαλοποιείται. Επομένως, μία ασφαλής και "οικονομική" επιλογή θα ήταν να σχεδίαζα σύστημα με μέσο ρυθμό εξυπηρέτησης μ=8 πελάτες/min.

δ) Στο 4ο διάγραμμα φαίνεται πως ανεξαρτήτως τον ρυθμό εξυπηρέτησης 'μ', η τιμή throughput είναι σταθερή και αμετάβλητη. Αυτό είναι λογικό, εφόσον η πιθανότητα απώλειας Pblocking είναι μηδενική λόγω της άπειρης χωρητικότητας της ουράς του συστήματος. Πιο συγκεκριμένα, η Ρυθμαπόδοση (throughput) υπολογίζεται μέσω του τύπου
$\gamma = \lambda(1 - P\text{blocking})$. Άρα στην περίπτωσή μας γ=λ=5.

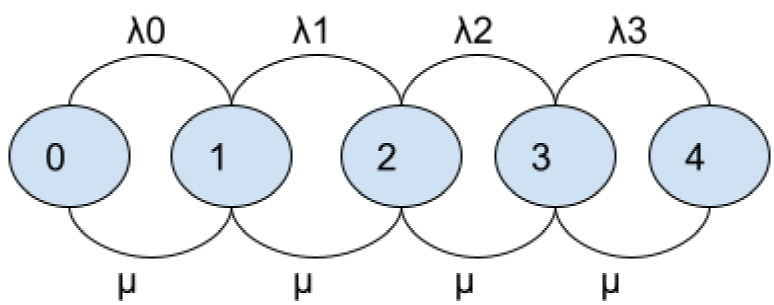
Ακολουθεί το printout του κώδικα octave:

```
#Queuing Systems | 6th Semester | NTUA ECE
#2nd Lab 2021
#2nd Exercise
#editor: Nikitas Tsinnas, 03118187
pkg load statistics
pkg load queueing

clc;
clear all;
close all;

#b
lamda = 5;
m = [6:1:10];
```

```
U=[0,length(m)]; #server utiliaztion
R=[0,length(m)]; #server responce time
Q=[0,length(m)]; #average number of requests in the system
X=[0,length(m)]; #server throughput
#Calculating U,R,Q,X with gsmm1 from queueing package
for i=1:length(m)
  [U(i),R(i),Q(i),X(i)] = qsmm1(lamda, m(i));
endfor

#Plotting Diagrams...
#Utiliaztion
figure(1);
plot (m,U);
title("Utiliaztion in respect to service rate 'm'");
ylabel("Utilaztion");
xlabel("'m'");
saveas(1, "figure1.jpg")
#responce time
figure(2);
plot (m,R);
title("responce time in respect to service rate 'm'");
ylabel("responce time");
xlabel("'m'");
saveas(2, "figure2.jpg")
#average number of requests
figure(3);
plot (m,Q);
title("average number of requests in respect to service rate 'm'");
ylabel("average number of requests");
xlabel("'m'");
saveas(3, "figure3.jpg")
#throughput
figure(4);
plot (m,X);
title("throughput in respect to service rate 'm'");
ylabel("throughput");
xlabel("'m'");
saveas(4, "figure4.jpg")
```

Διαδικασία γεννήσεων θανάτων (birth-death process): εφαρμογή σε σύστημα M/M/1/K

α)



Έχουμε:

- $\lambda_0 P_0 = \mu_1 P_1 \Rightarrow \lambda_{(k-1)} * P_{(k-1)} = \mu_k P_k$, k = 1, 2, 3, 4
- $P_0 + P_1 + P_2 + P_3 + P_4 = 1$
- $Pk = [\lambda^k/(k!\,\mu^\kappa)]P0$, k = 1, 2, 3, 4
- Εργοδικές πιθανότητες:
  P0 = 0.60664, P1 = 0.30332, P2 = 0.075830, P3 = 0.012638
  P4 = 0.0015798
- Πιθανότητα απώλειας πελάτη:
  Pblocking = P4 = 0.0015798

β)

i) πίνακας μεταβάσεων:

-5.0000	5.0000	0	0	0
10.0000	-12.5000	2.5000	0	0
0	10.0000	-11.6667	1.6667	0
0	0	10.0000	-11.2500	1.2500
0	0	0	10.0000	-10.0000

ii) Οι εργοδικές πιθανότητες που υπολογίζονται μέσω octave ταυτίζονται με αυτές που υπολογίστηκαν παραπάνω

P =

6.0664e-01   3.0332e-01   7.5829e-02   1.2638e-02   1.5798e-03

iii) Ο μέσος αριθμός πελατών στο σύστημα όταν αυτό ισορροπεί υπολογίζεται μέσω octave:

```
 Average Number of customers in the system when on equilibrium :
0.4992
```

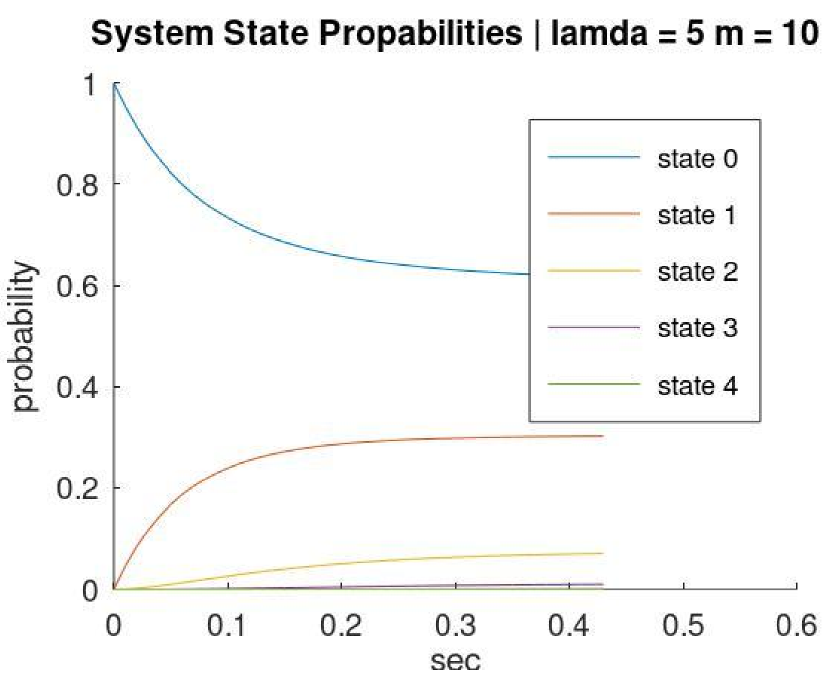
Δηλαδή, 1*P1 + 2*P2 + 3*P3 + 4*P4 = 0.4992.

iv) Υπολογίζεται η πιθανότητα απόρριψης πελάτη:

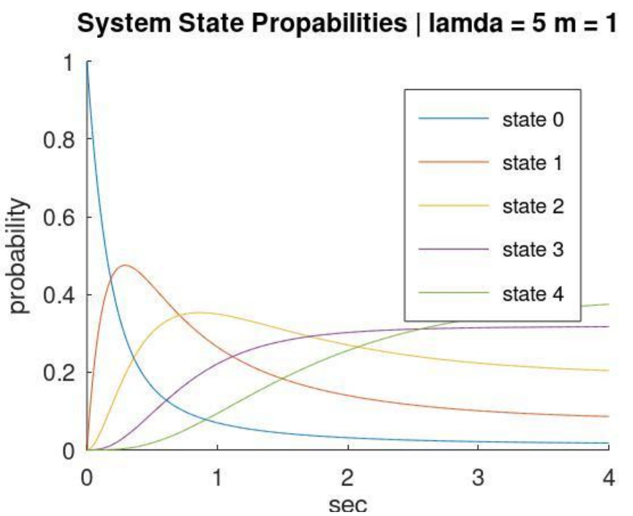
```
 Probability of losing a customer :
1.5798e-03
```

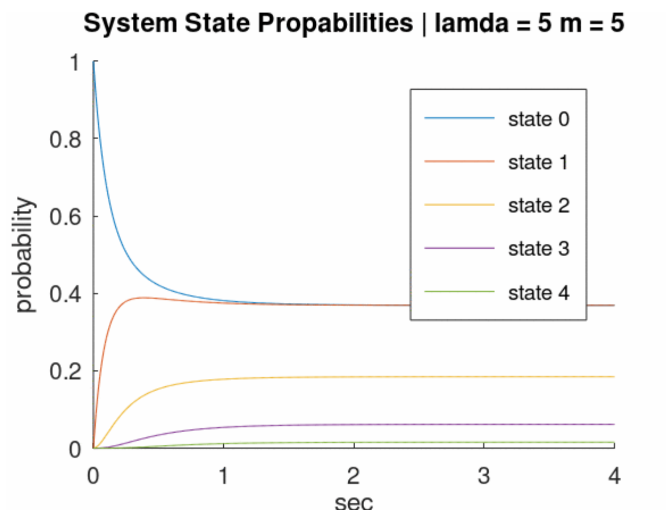
τιμή η οποία επιβεβαιώνει τους προηγούμενους υπολογισμούς.

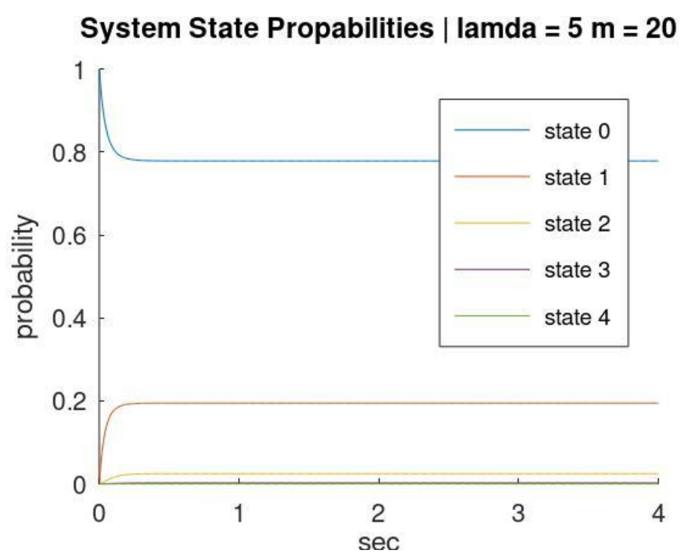
v)



vi)







Παρατηρούμε πως όσο μικραίνει το ρ=λ/μ, τόσο τείνει η πιθανότητα της αρχικής κατάστασης P0 στο 1 και οι πιθανότητες των υπόλοιπων καταστάσεων στο 0, ενώ στον χρόνο παραμένουν σχεδόν σταθερές.. Αυτό είναι λογικό, εφόσον μικρό ρ σημαίνει πως ο ρυθμός άφιξης ανά τον ρυθμό εξυπηρέτησης είναι πολύ μικρός και άρα δεν υπάρχουν θεωρητικά πελάτες που περιμένουν στο σύστημα εφόσον εξυπηρετούνται πολύ γρήγορα κατά την άφιξή τους.
Για ρ=1 παρατηρούμε πως οριακά οι πιθανότητες της κατάστασης 0 και 1 σχεδόν ταυτίζονται και δεν μεταβάλλονται, δηλαδή έχουν σταθερή κλίση.

Για ρ=5>1 βλέπουμε πως μετά την πάροδο των 5sec η πιο πιθανή κατάσταση είναι αυτή της απόρριψης πελάτη. Κατά την πάροδο του χρόνου παρατηρούμε πως παρουσιάζουν μέγιστο σειριακά οι καταστάσεις του συστήματος (δηλαδή πρώτα η 1η, μετά η 2η κτλ) έως τα 5 sec όπου φαίνεται το σύστημα να έχει ισορροπήσει πιθανοτικά στην τελευταία κατάσταση, η οποία είναι και μη επιθυμητή.

Ακολουθεί το printout του κώδικα Octave:

```
#Queuing Systems | 6th Semester | NTUA ECE
#2nd Lab 2021
#3rd Exercise
#editor: Nikitas Tsinnas, 03118187
pkg load statistics
pkg load queueing

clc;
clear all;
close all;

#bi
states = [0, 1, 2, 3, 4];
lamda = 5;
m = 10;
% birth rate = arrival rate (poisson parameter)
births = [lamda, lamda/2, lamda/3, lamda/4];
# death rate = service rate (exponential parameter)
deaths = [m, m, m, m];
% calculate and print transition table of process
display (ctmcbd(births, deaths))

#bii
transition_table = ctmcbd(births, deaths)
P = ctmc(transition_table);
display (P);
#biii
display("Average number of customers in the system when on equilibrium:")
display(sum(P.*[0,1,2,3,4]))

#biv
display("Probability of blocking customer:")
display(P(5))

#bv
% plot the ergodic probabilities (bar for bar chart)
i = 0;
```

```
initial_state = [1, 0, 0, 0, 0];
for T = 0 : 0.01 : 100
  i = i + 1;
  P0 = ctmc(transition_table, T, initial_state);
  p0(i) = P0(1);
  p1(i) = P0(2);
  p2(i) = P0(3);
  p3(i) = P0(4);
  p4(i) = P0(5);
  if P0 - P < 0.01
    break;
  endif
endfor

T = 0 : 0.01 : T; #make new time vector
figure(1);
title(strjoin({"System State Propabilities | lamda = ",num2str(lamda)," m =
",num2str(m)},""))
xlabel("sec")
ylabel("probability")
hold on;
plot(T, p0);
plot(T, p1);
plot(T, p2);
plot(T, p3);
plot(T, p4);
legend("state 0","state 1","state 2","state 3","state 4");
hold off;
%bvi
m = [1,5,20];
for i=1:columns(m)
  deaths = [m(i), m(i), m(i), m(i)];
  transition_table = ctmcbd(births, deaths);
  idx = 0;
  for T = 0 : 0.01 : 4
    idx = idx + 1;
    P0 = ctmc(transition_table, T, initial_state);
    p0(idx) = P0(1);
    p1(idx) = P0(2);
    p2(idx) = P0(3);
    p3(idx) = P0(4);
    p4(idx) = P0(5);
    if P0 - P < 0.01
      break;
    endif
  endfor

  T = 0 : 0.01 : T;
```

```
  figure(i+1);
  title(strjoin({"System State Propabilities | lamda = ",num2str(lamda)," m = 
",num2str(m(i))},""))
  xlabel("sec")
  ylabel("probability")
  hold on;
  plot(T, p0);
  plot(T, p1);
  plot(T, p2);
  plot(T, p3);
  plot(T, p4);
  legend("state 0","state 1","state 2","state 3","state 4");
  hold off;
endfor
```