ΠΛΗ30

ΕΝΟΤΗΤΑ 2: ΣΧΕΔΙΑΣΗ ΑΛΓΟΡΙΘΜΩΝ

Μάθημα 2.3:
Άπληστοι Αλγόριθμοι

Δημήτρης Ψούνης

www.psounis.gr

ΠΕΡΙΕΧΟΜΕΝΑ

Α. Σκοπός του Μαθήματος

Οι στόχοι του μαθήματος είναι:

Επίπεδο Α

- (-)

Επίπεδο Β

- Η τεχνική σχεδίασης αλγόριθμων Άπληστος Αλγόριθμος
- Ο άπληστος αλγόριθμος για την ελαχιστοποίηση νομισμάτων για ρέστα.

Επίπεδο Γ

- Ο άπληστος αλγόριθμος του Dijkstra για την εύρεση του συντομότερου μονοπατιού
- Ο άπληστος αλγόριθμος του Prim για την εύρεση Ελάχιστου Συνδετικού Δένδρου
- Ο άπληστος αλγόριθμος του Kruskal για την εύρεση Ελάχιστου Συνδετικού Δένδρου

Β. Θεωρία
Τεχνικές Σχεδίασης Αλγορίθμων

- Στην 2η ενότητα του μαθήματος ασχολούμαστε με τεχνικές που έχουν αναπτυχθεί, ως γενικές μεθοδολογίες για την κατασκευή ενός αλγορίθμου:
  - Η τεχνική Διαίρει και Βασίλευε (Μάθημα 2.1)
  - Η τεχνική του Δυναμικού Προγραμματισμού (Μάθημα 2.2)
  - **Η κατασκευή των Άπληστων Αλγόριθμων (Μάθημα 2.3)**
- Υπάρχουν ακόμη δεκάδες τεχνικές κατασκευής αλγορίθμων που είναι εκτός ύλης.

Β. Θεωρία
1. Άπληστοι Αλγόριθμοι

Όταν έχουμε ένα πρόβλημα που έχει τις εξής ιδιότητες:

- Ιδιότητα της Άπληστης Επιλογής: Μια ακολουθία άπληστων επιλογών οδηγεί στην βέλτιστη λύση.
- Ιδιότητα των Βέλτιστων Επιμέρους Δομών: Οτί για να λύσουμε το πρόβλημα αρκεί να υπολογίσουμε την βέλτιστη λύση σε κάποια υποπροβλήματα, συνήθως με αναδρομή.

Τότε επιλέγουμε ως τεχνική σχεδίασης κατασκευής άπληστου αλγόριθμου

Ένας άπληστος αλγόριθμος δεν είναι πάντα βέλτιστος:

- Για να δείξουμε ότι δεν είναι βέλτιστος δίνουμε ένα αντιπαράδειγμα.
- Για να δείξουμε ότι είναι βέλτιστος δίνουμε απόδειξη ορθότητας:
  1. Θεωρούμε ότι ο άπληστος αλγόριθμος δεν επιστρέφει την βέλτιστη λύση.
  2. Απεικονίζουμε την βέλτιστη λύση και την λύση του άπληστου αλγόριθμου.
  3. Συγκρίνουμε τις λύσεις με βάση το άπληστο κριτήριο.
  4. Δείχνουμε ότι η λύση του άπληστου αλγόριθμου στο σημείο που διαφέρουν, είναι καλύτερη από την βέλτιστη λύση.

Β. Θεωρία

1. Άπληστοι Αλγόριθμοι

1. Συντομότερο Μονοπάτι σε Γράφο

- ΠΡΟΒΛΗΜΑ: Δίνεται γράφος G=(V,E,W), αφετηρία s∈V, τερματισμός t ∈V. Ζητείται το συντομότερο μονοπάτι από την s στην t.

- Θα μελετήσουμε έναν αλγόριθμο που υπολογίζει το συντομότερο μονοπάτι:
  - Ο αλγόριθμος του Dijkstra:
    - Θεωρεί (άπληστα?) ότι σε κάθε επανάληψη βρίσκει το συντομότερο μονοπάτι για να πάει από την αφετηρία σε μία κορυφή
      - Συγκεκριμένα την κορυφή που απέχει λιγότερο από την αφετηρία στο τρέχων βήμα.

Β. Θεωρία

1. Άπληστοι Αλγόριθμοι

1. Συντομότερο Μονοπάτι σε Γράφο

Σκιαγράφηση Αλγόριθμου Dijkstra:

- Στην αρχικοποίηση:
  - Θέτουμε όλες τις ετικέτες L[.]=+∞ εκτός της αφετηρίας που έχει L[s]=0
- Σε κάθε βήμα:
  - Οριστικοποιείται η κορυφή με το μικρότερο κόστος από τις μη οριστικοποιημένες
  - Διορθώνονται οι ετικέτες των γειτονικών μη οριστικοποιημένων κορυφών (σε περίπτωση που βρεθεί καλύτερο μονοπάτι από την κορυφή που οριστικοποιήθηκε)
- Τερματισμός:
  - Όταν οριστικοποιηθεί η κορυφή τερματισμού t.

Β. Θεωρία

1. Άπληστοι Αλγόριθμοι

1. Συντομότερο Μονοπάτι σε Γράφο (Dijkstra-Ψευδοκώδικας)

```
procedure Dijkstra(G=(V,E,W), s, t)
   L[s]=0
   T=V
   for all x∈V-{s}
      L[x]=+∞
   end for

   while t∈T do
      Επέλεξε v∈T με ελάχιστο L[v]
      T=T-{v}
      for all x∈T γειτονική της v:
         if (L[v]+W[v,x]<L[x])
            L[x]=L[v]+W[v,x]
            P[x]=v
         end if
     end for
   end while

   return L[t]

end procedure
```

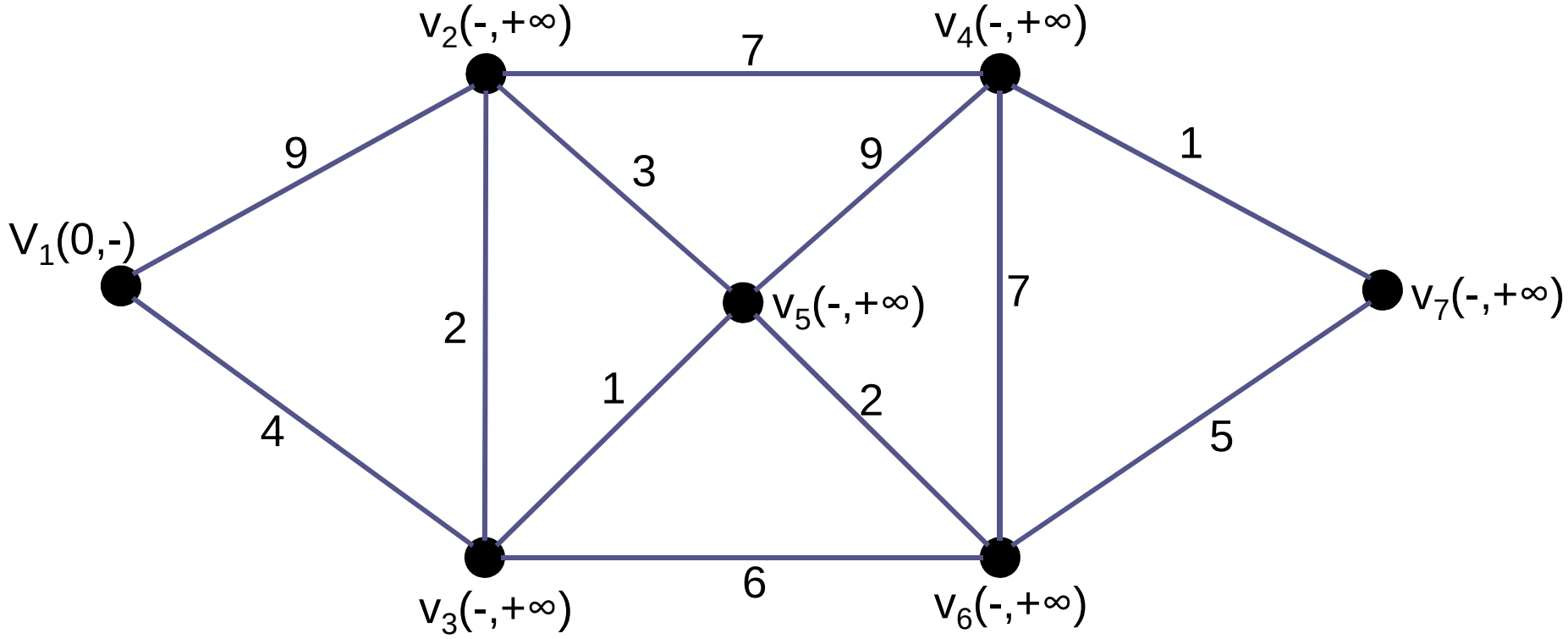
Β. Θεωρία

1. Άπληστοι Αλγόριθμοι

1. Συντομότερο Μονοπάτι σε Γράφο(Dijkstra-Παράδειγμα Εκτέλεσης)

Ψάχνουμε το συντομότερο μονοπάτι από την v_1 στην v_7

- Βήμα 0:



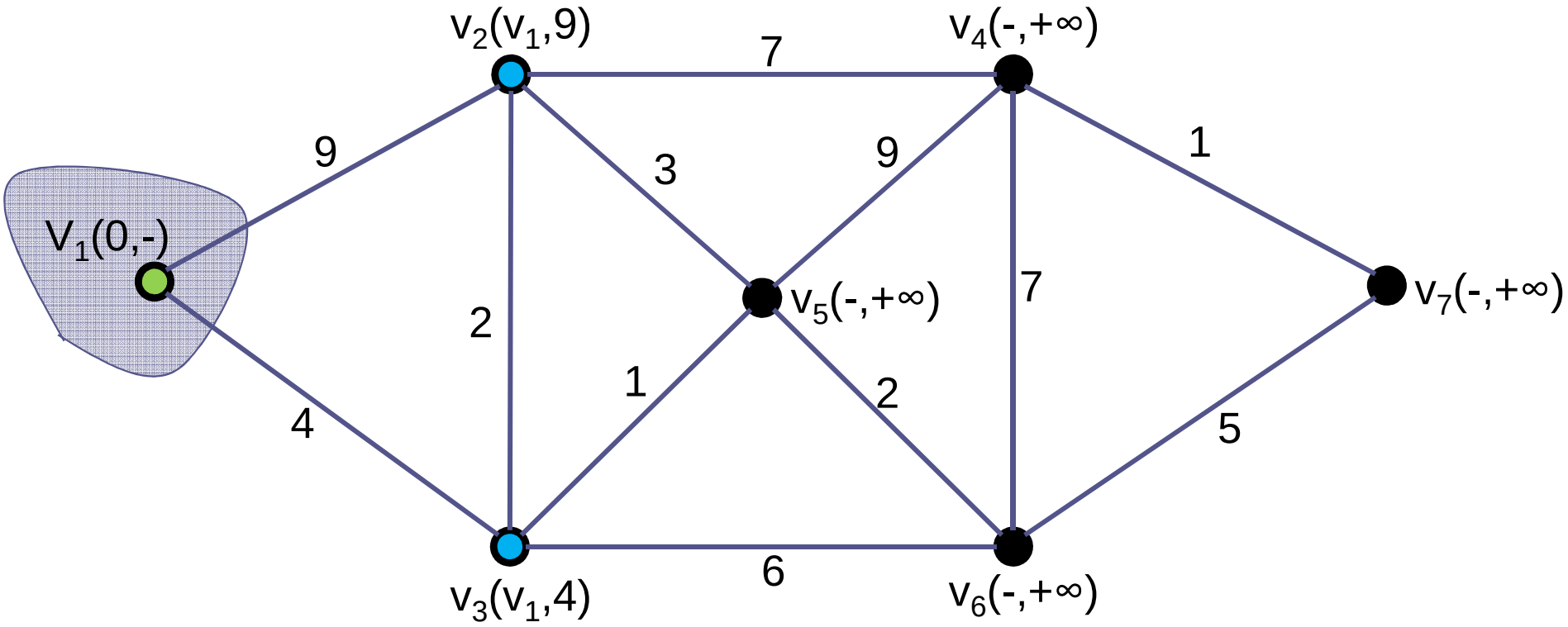
Αρχικοποίηση ετικετών κορυφών

Β. Θεωρία

1. Άπληστοι Αλγόριθμοι

1. Συντομότερο Μονοπάτι σε Γράφο(Dijkstra-Παράδειγμα Εκτέλεσης)

- Βήμα 1:



Οριστικοποίηση κορυφής v_1

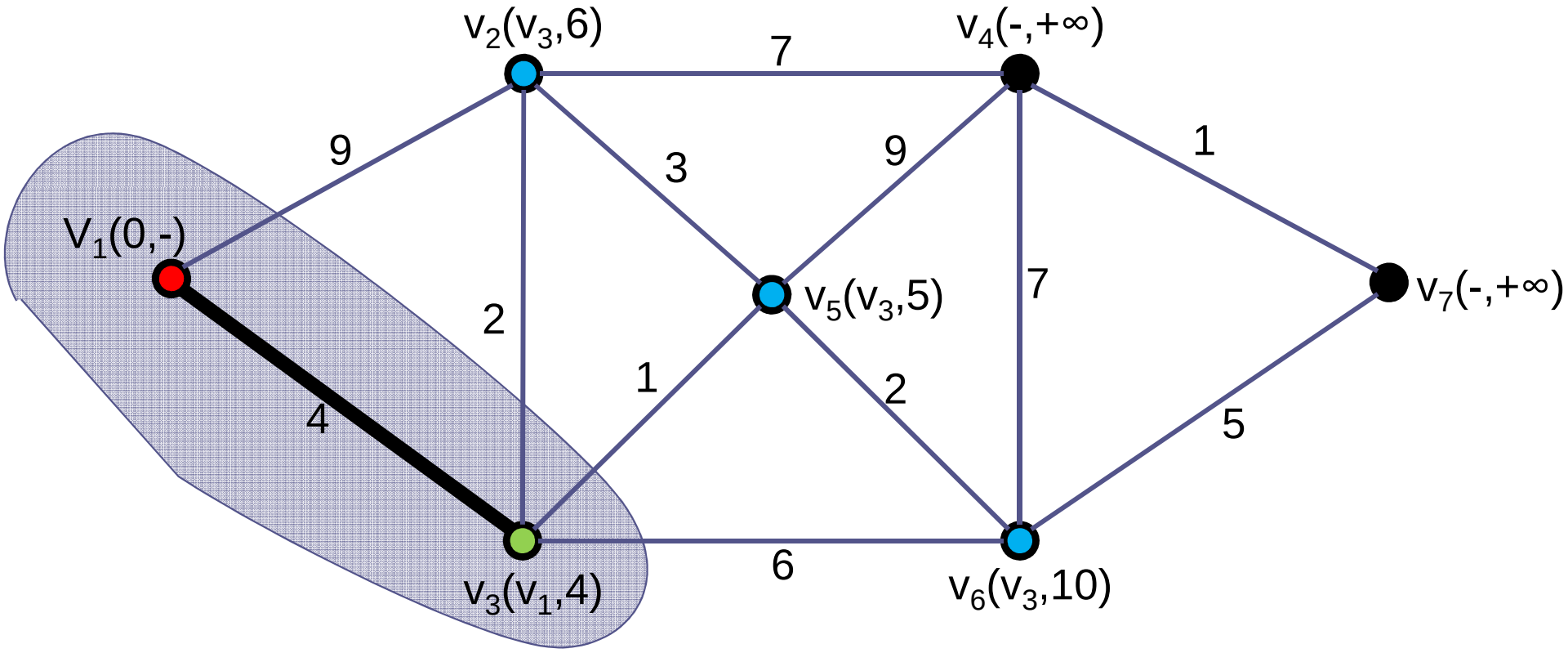
Εξεταση κορυφών v_2,v_3. Διόρθωση ετικετών v_2,v_3

Β. Θεωρία

1. Άπληστοι Αλγόριθμοι

1. Συντομότερο Μονοπάτι σε Γράφο(Dijkstra-Παράδειγμα Εκτέλεσης)

- Βήμα 2:



Οριστικοποίηση κορυφής v_3

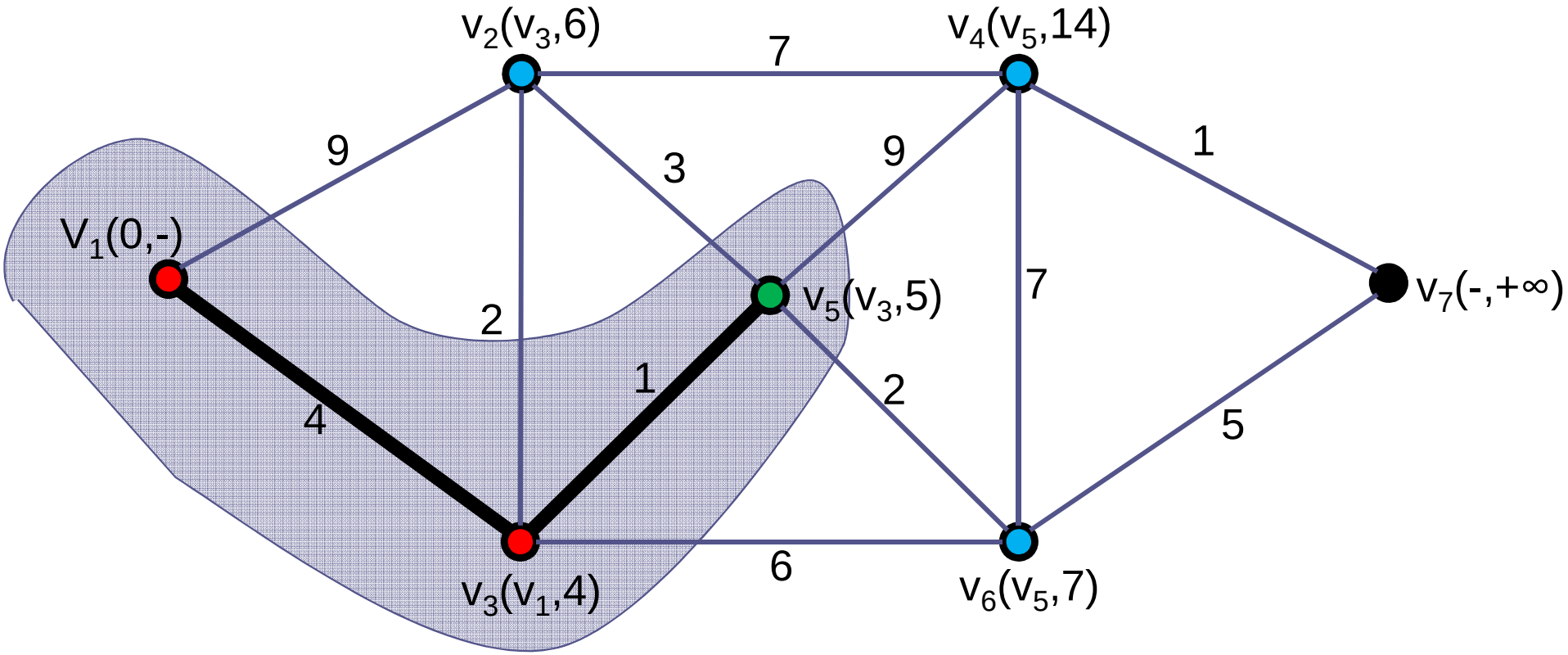
Εξεταση κορυφών v_2,v_5,v_6. Διόρθωση ετικετών v_2,v_5,v_6

Β. Θεωρία

1. Άπληστοι Αλγόριθμοι

1. Συντομότερο Μονοπάτι σε Γράφο(Dijkstra-Παράδειγμα Εκτέλεσης)

- Βήμα 3:



Οριστικοποίηση κορυφής v_5

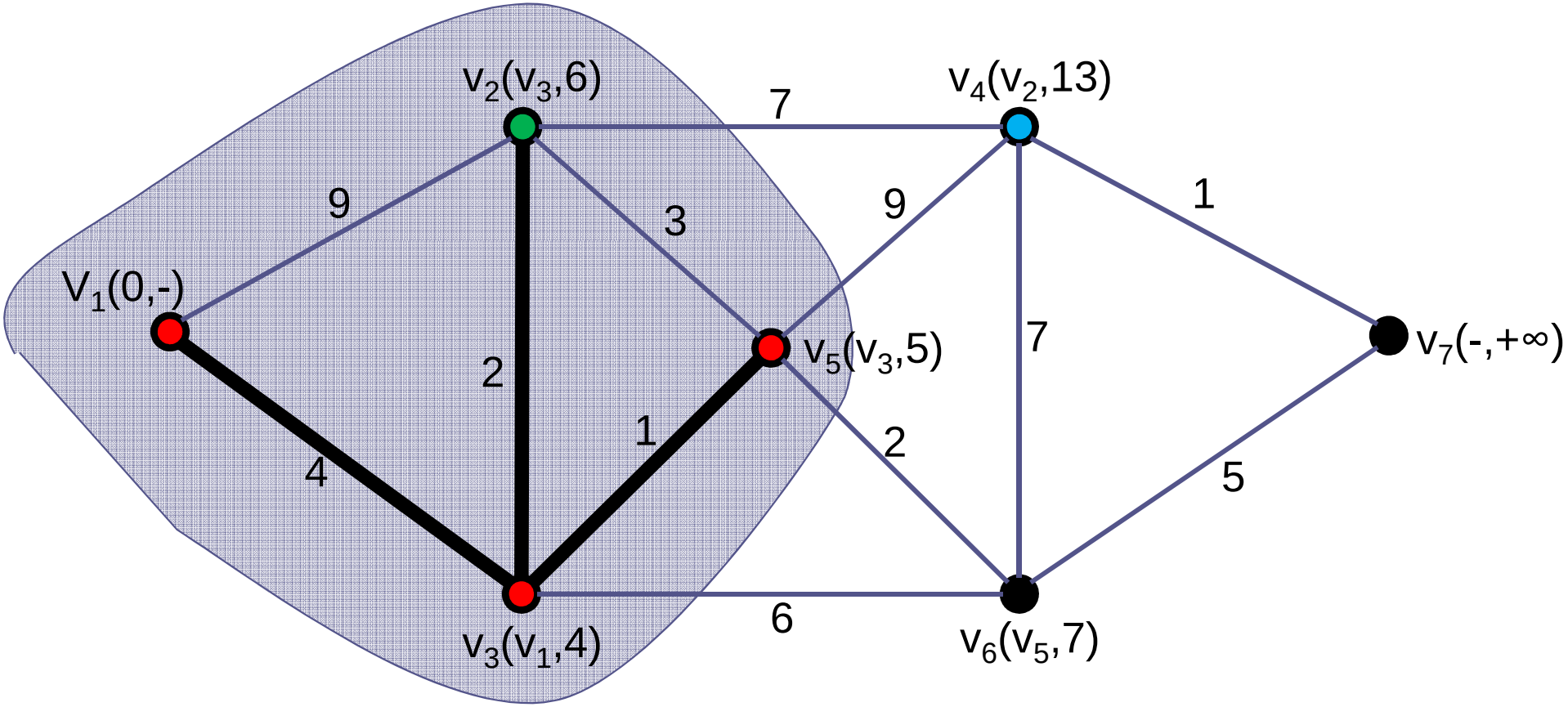
Εξεταση κορυφών v_2,v_4,v_6. Διόρθωση ετικετών v_4,v_6

Β. Θεωρία

1. Άπληστοι Αλγόριθμοι

1. Συντομότερο Μονοπάτι σε Γράφο(Dijkstra-Παράδειγμα Εκτέλεσης)

- Βήμα 4:



Οριστικοποίηση κορυφής v_2

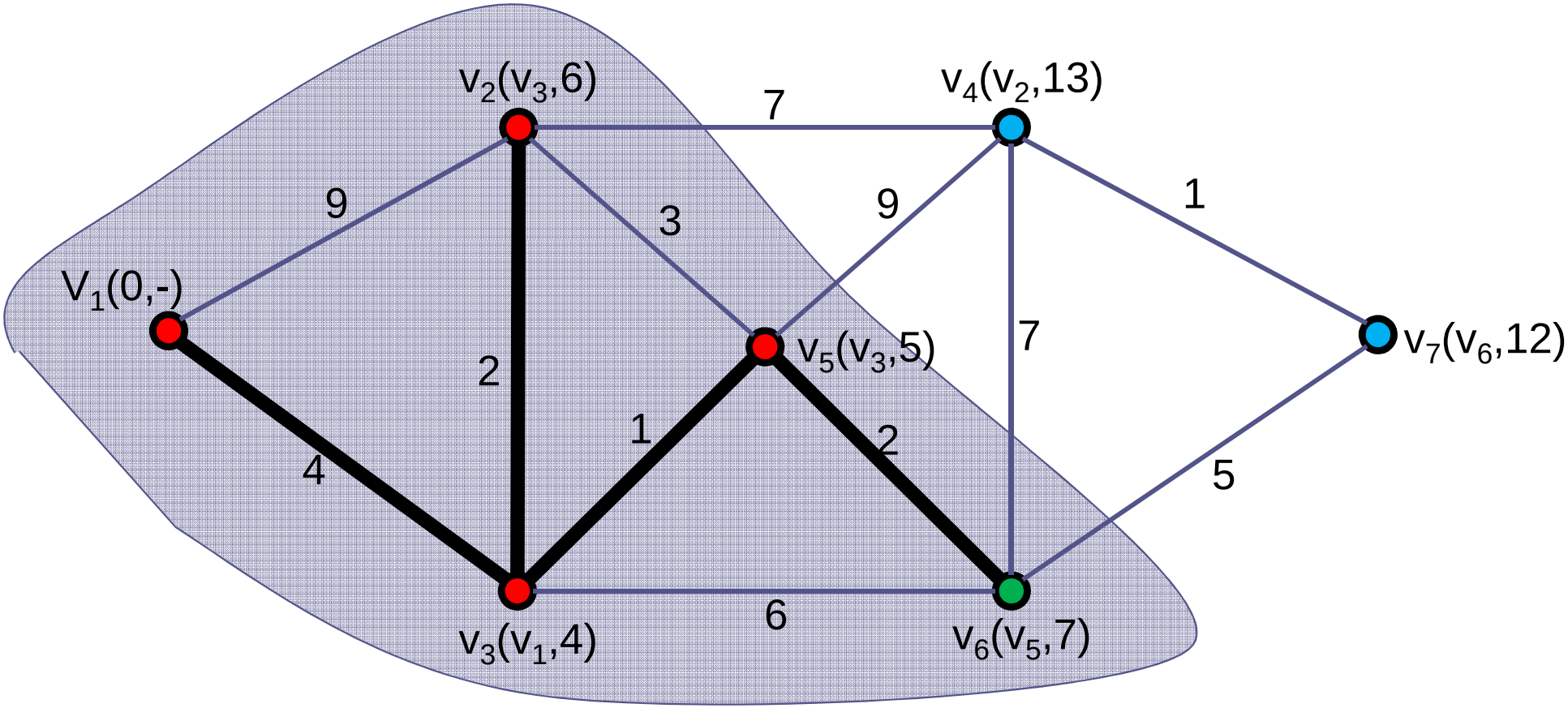
Εξεταση κορυφής v_4. Διόρθωση ετικέτας v_4

Β. Θεωρία

1. Άπληστοι Αλγόριθμοι

1. Συντομότερο Μονοπάτι σε Γράφο(Dijkstra-Παράδειγμα Εκτέλεσης)

- Βήμα 5:



Οριστικοποίηση κορυφής v_6

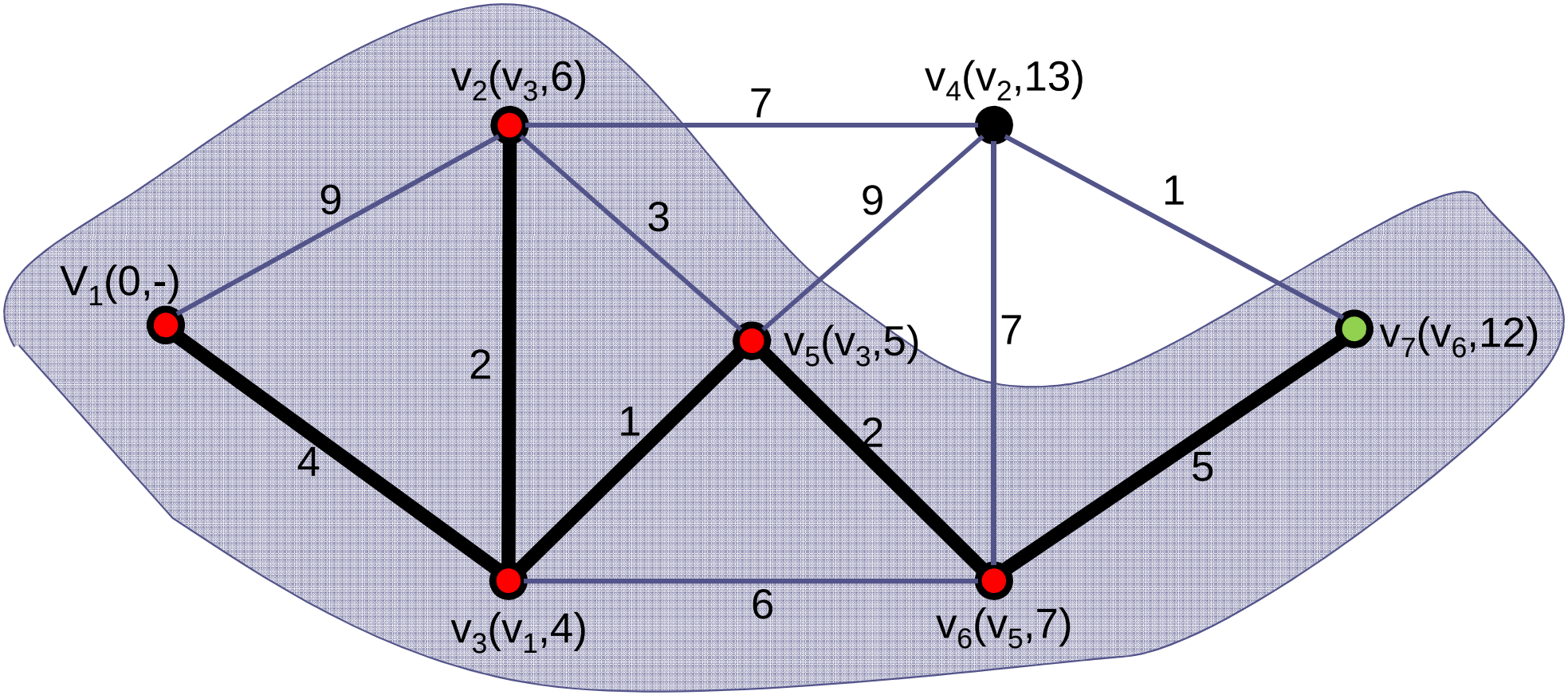
Εξεταση κορυφών v_4,v_7. Διόρθωση ετικετας v_7

Β. Θεωρία

1. Άπληστοι Αλγόριθμοι

1. Συντομότερο Μονοπάτι σε Γράφο(Dijkstra-Παράδειγμα Εκτέλεσης)

- Βήμα 6:



Οριστικοποίηση κορυφής v_7

Τέλος Αλγορίθμου. Συντομότερο μονοπάτι v_1-v_3-v_5-v_6-v_7 με βάρος 4+1+2+5=12

Β. Θεωρία

1. Άπληστοι Αλγόριθμοι

1. Συντομότερο Μονοπάτι σε Γράφο (Dijkstra – Απόδειξη Ορθότητας)

Απόδειξη Ορθότητας αλγορίθμου Dijkstra:

- Θεωρούμε ότι η λύση που υπολογίζει ο άπληστος αλγόριθμος δεν είναι βέλτιστη και
- Συμβολίζουμε με:
  - OPT η βέλτιστη λύση (ακολουθία κορυφών από την s στην t)
  - C η λύση που επιστρέφει ο αλγόριθμος του Dijkstra.
- Κάθε μονοπάτι που υπολογίζει ο Dijkstra είναι βέλτιστος
  - Πράγματι, η άπληστη επιλογή του Dijkstra είναι ορθή για την οριστικοποίηση της κορυφής v_i
  - Αν δεν ήταν θα υπήρχε ένα άλλο μονοπάτι για να φτάσω στην v_i με μικρότερο κόστος.
  - Άτοπο, γιατί όταν οριστικοποιείται η v_i έχει το μικρότερο κόστος από όλες τις υπόλοιπες κορυφές.

Β. Θεωρία

1. Άπληστοι Αλγόριθμοι

2. Ελάχιστο Συνδετικό Δένδρο

- ΠΡΟΒΛΗΜΑ: Δίνεται γράφος G=(V,E,W), Ζητείται ένα ελάχιστο συνδετικό δένδρο (δένδρο που περιλαμβάνει όλες τις κορυφές και έχει ελάχιστο βάρος

- Θα μελετήσουμε δύο αλγόριθμους που υπολογίζουν ελάχιστα συνδετικά δένδρα:
  - Ο αλγόριθμος του Prim:
    - Θεωρεί (άπληστα?) ότι η επιλογή ακμής που έχει το ένα της άκρο στο συνδετικό δένδρο και το άλλο άκρο της εκτός του συνδετικού δένδρου θα οδηγήσει στην βέλτιστη λύση
  - Ο αλγόριθμος του Kruskal:
    - Θεωρεί (άπληστα?) ότι η ακμή ελαχίστου βάρους που δεν βρίσκεται στο δένδρο και δεν δημιουργεί κύκλο θα μπορούσε να ενσωματωθεί στην λύση ώστε να κατασκευασθεί το ελάχιστο συνδετικό δένδρο.

Β. Θεωρία

1. Άπληστοι Αλγόριθμοι

2. Ελάχιστο Συνδετικό Δένδρο (1.Prim-Σκιαγράφηση)

Σκιαγράφηση Αλγόριθμου Prim:

- Στην αρχικοποίηση:
  - Τοποθετούμε αυθαίρετα μια κορυφή στο συνδετικό δένδρο
- Σε κάθε βήμα:
  - Υποψήφιες ακμές για να μπουν στο συνδετικό δένδρο είναι εκείνες οι ακμές που έχουν το ένα τους άκρο στο υπο κατασκευή συνδετικό δένδρο και το άλλο τους άκρο εκτός του συνδετικού δένδρου.
  - Επιλέγεται η ακμή με το ελάχιστο βάρος από τις υποψήφιες
  - Η ακμή εισάγεται στο δένδρο καθώς και το άκρο της που δεν ανήκε στο δένδρο.
- Τερματισμός:
  - Όταν όλες οι κορυφές εισαχθούν στο δένδρο.

Β. Θεωρία

1. Άπληστοι Αλγόριθμοι

2. Ελάχιστο Συνδετικό Δένδρο (1.Prim-Ψευδοκώδικας)

```
procedure Prim(G=(V,E,W))

   V'={v1}

   while |V'|<|V| do
      Βρες την ακμή (v,w)∈E με v∈V', w∈V-V' με το ελάχιστο βάρος
      Θέσε (v,w) στο E' και w στο V'
   end while

   return T=(V',E')

end procedure
```

Αποδεικνύεται ότι η πολυπλοκότητα του Prim με χρήση κατάλληλων δομών δεδομένων είναι $O(n^2)$

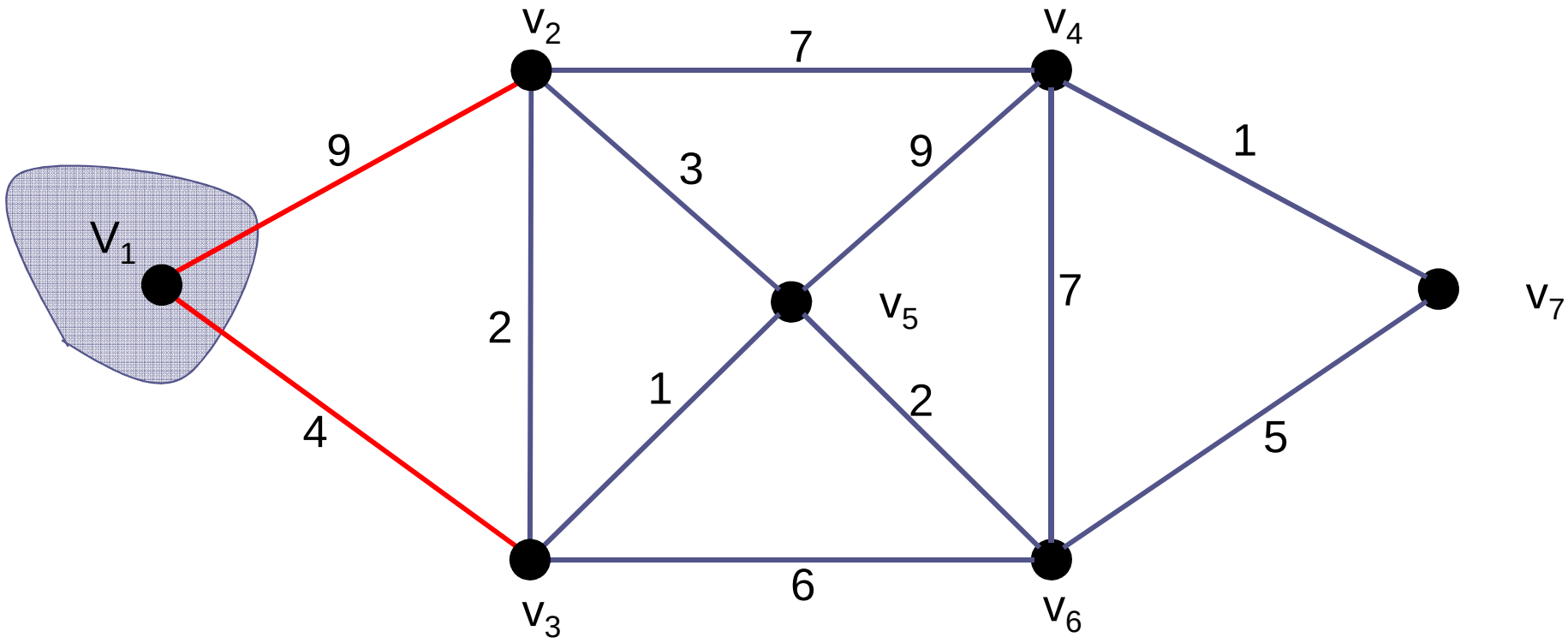
Β. Θεωρία

1. Άπληστοι Αλγόριθμοι

2. Ελάχιστο Συνδετικό Δένδρο (1.Prim-Παράδειγμα Εκτέλεσης)

Ξεκινάμε αυθαίρετα από την κορυφή v_1

- Βήμα 0:



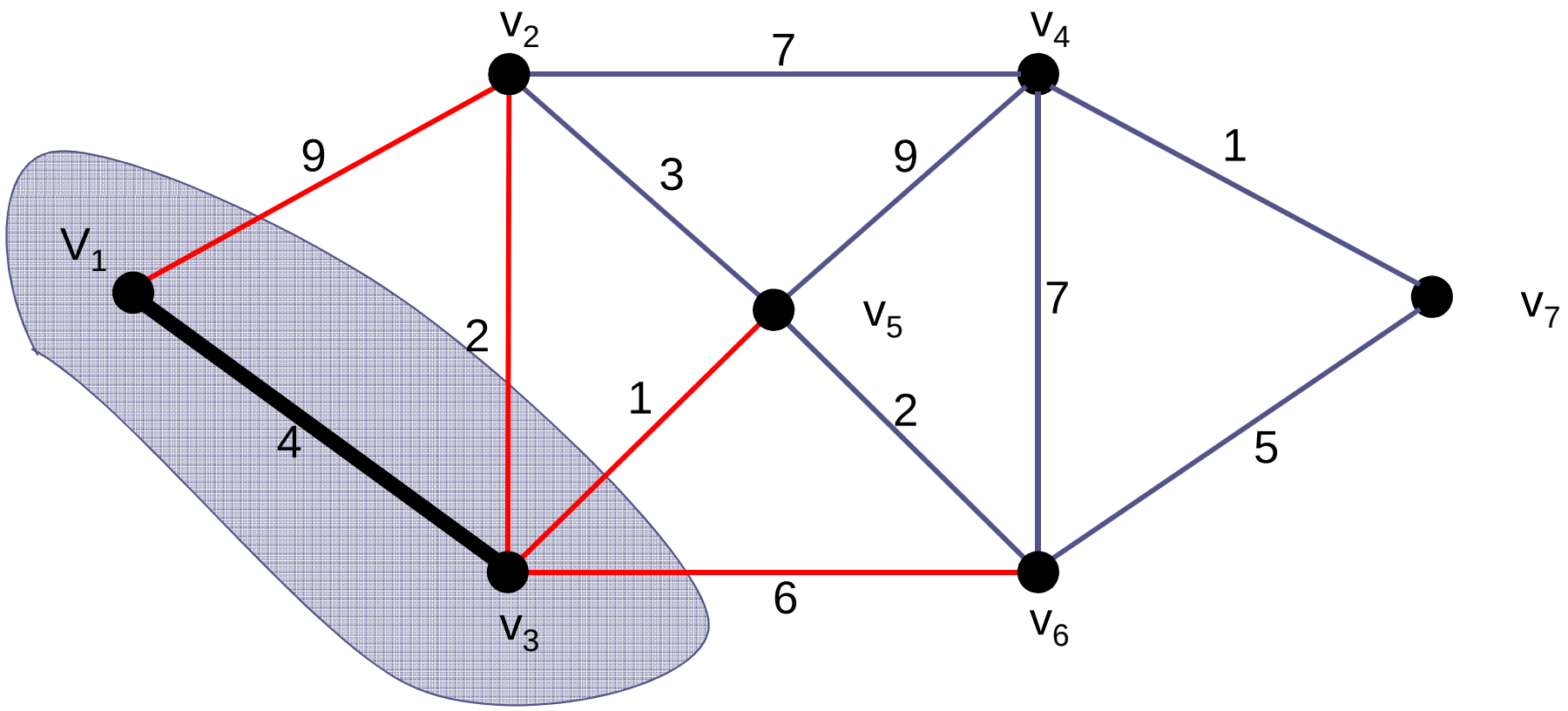
V_1 μπαίνει στο δένδρο

Β. Θεωρία

1. Άπληστοι Αλγόριθμοι

2. Ελάχιστο Συνδετικό Δένδρο (1.Prim-Παράδειγμα Εκτέλεσης)

- Βήμα 1:



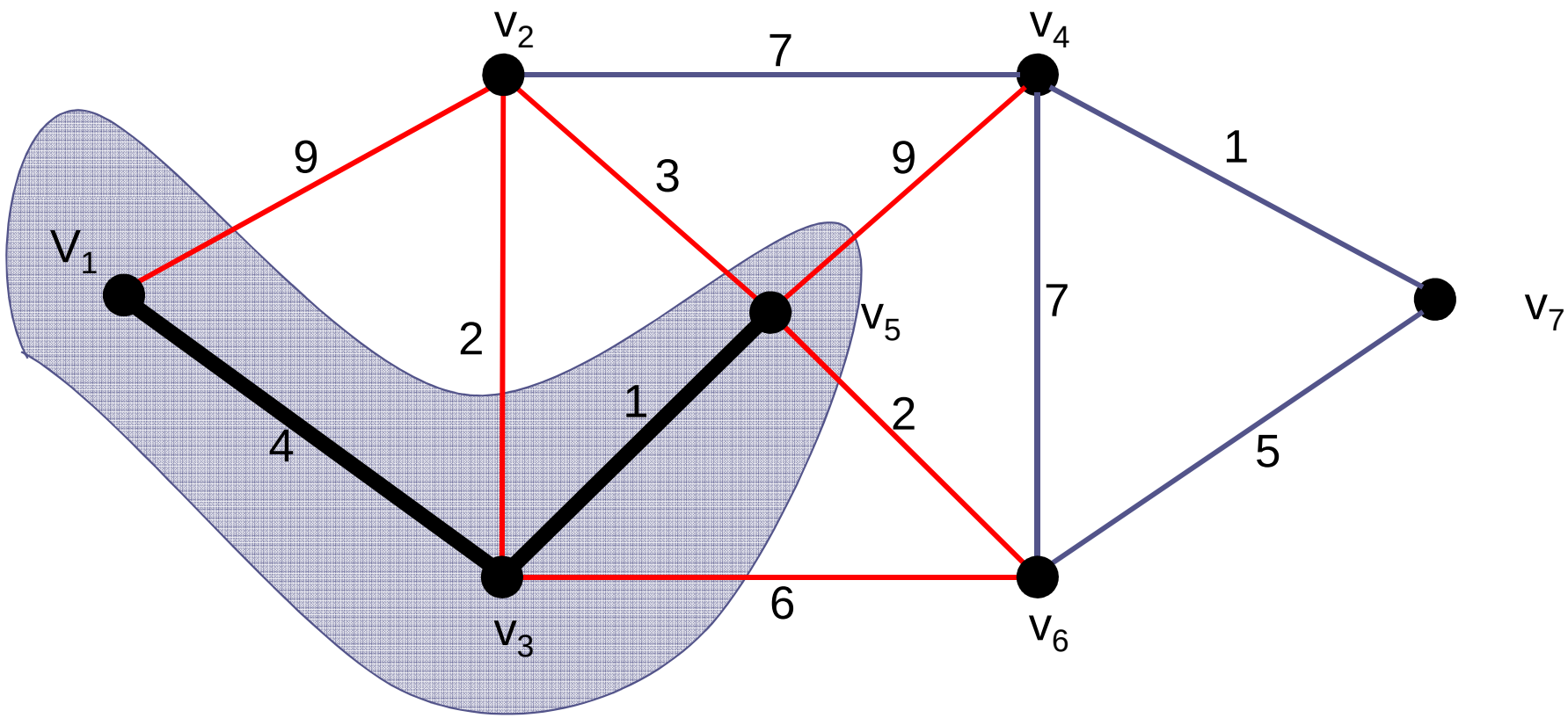
V_3 μπαίνει στο δένδρο

Β. Θεωρία

1. Άπληστοι Αλγόριθμοι

2. Ελάχιστο Συνδετικό Δένδρο (1.Prim-Παράδειγμα Εκτέλεσης)

- Βήμα 2:



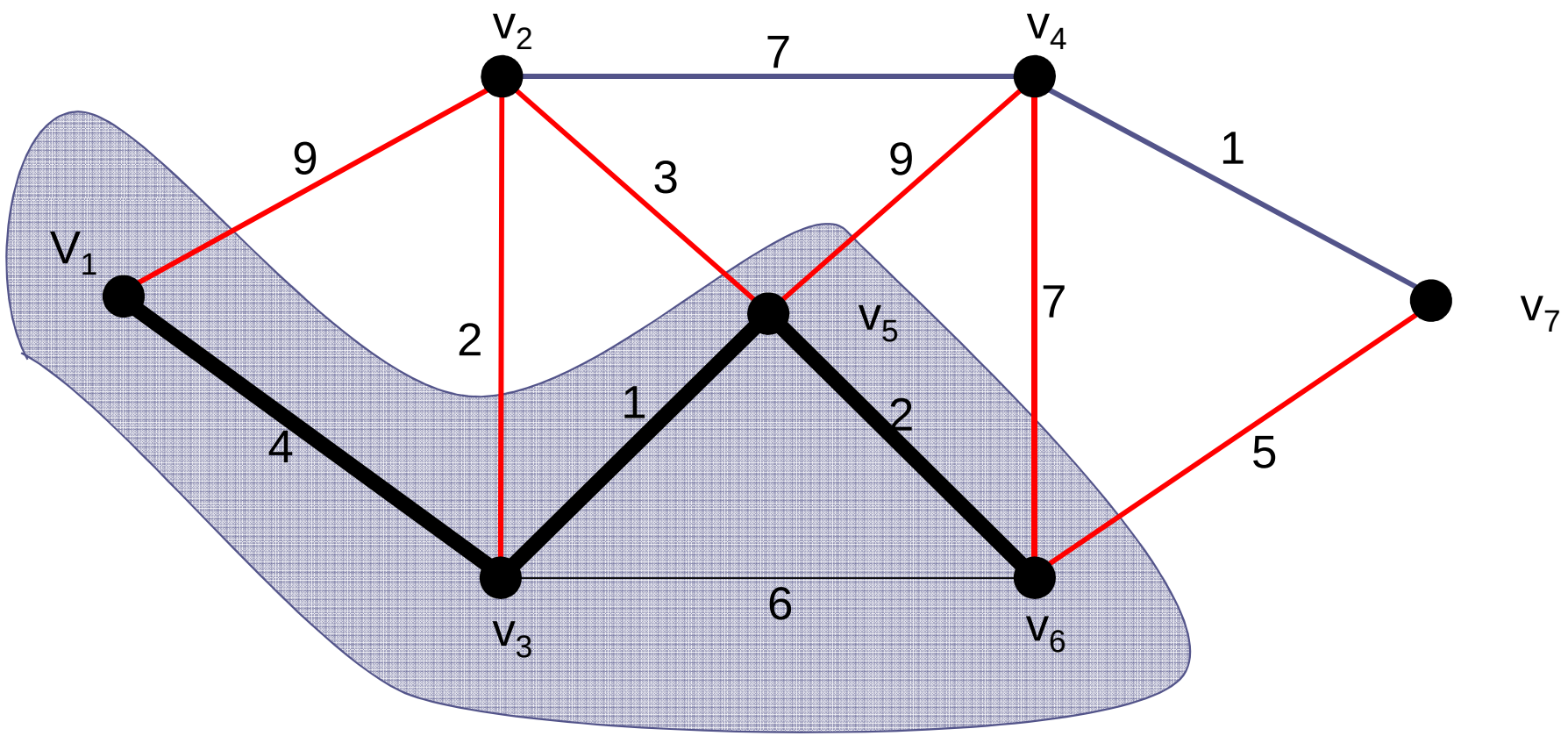
V_5 μπαίνει στο δένδρο

Β. Θεωρία

1. Άπληστοι Αλγόριθμοι

2. Ελάχιστο Συνδετικό Δένδρο (1.Prim-Παράδειγμα Εκτέλεσης)

- Βήμα 3:



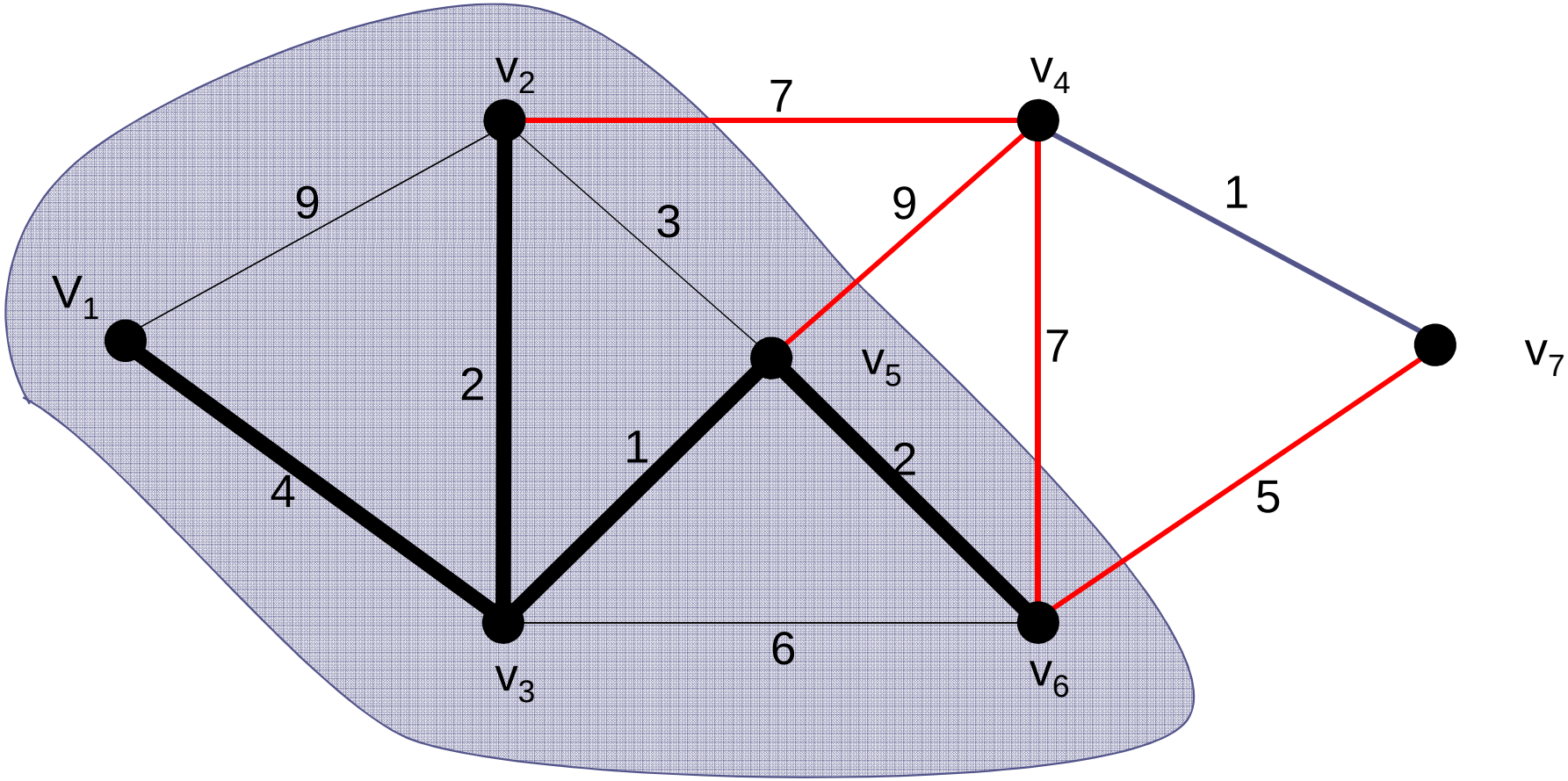
V_6 μπαίνει στο δένδρο

Β. Θεωρία

1. Άπληστοι Αλγόριθμοι

2. Ελάχιστο Συνδετικό Δένδρο (1.Prim-Παράδειγμα Εκτέλεσης)

- Βήμα 4:



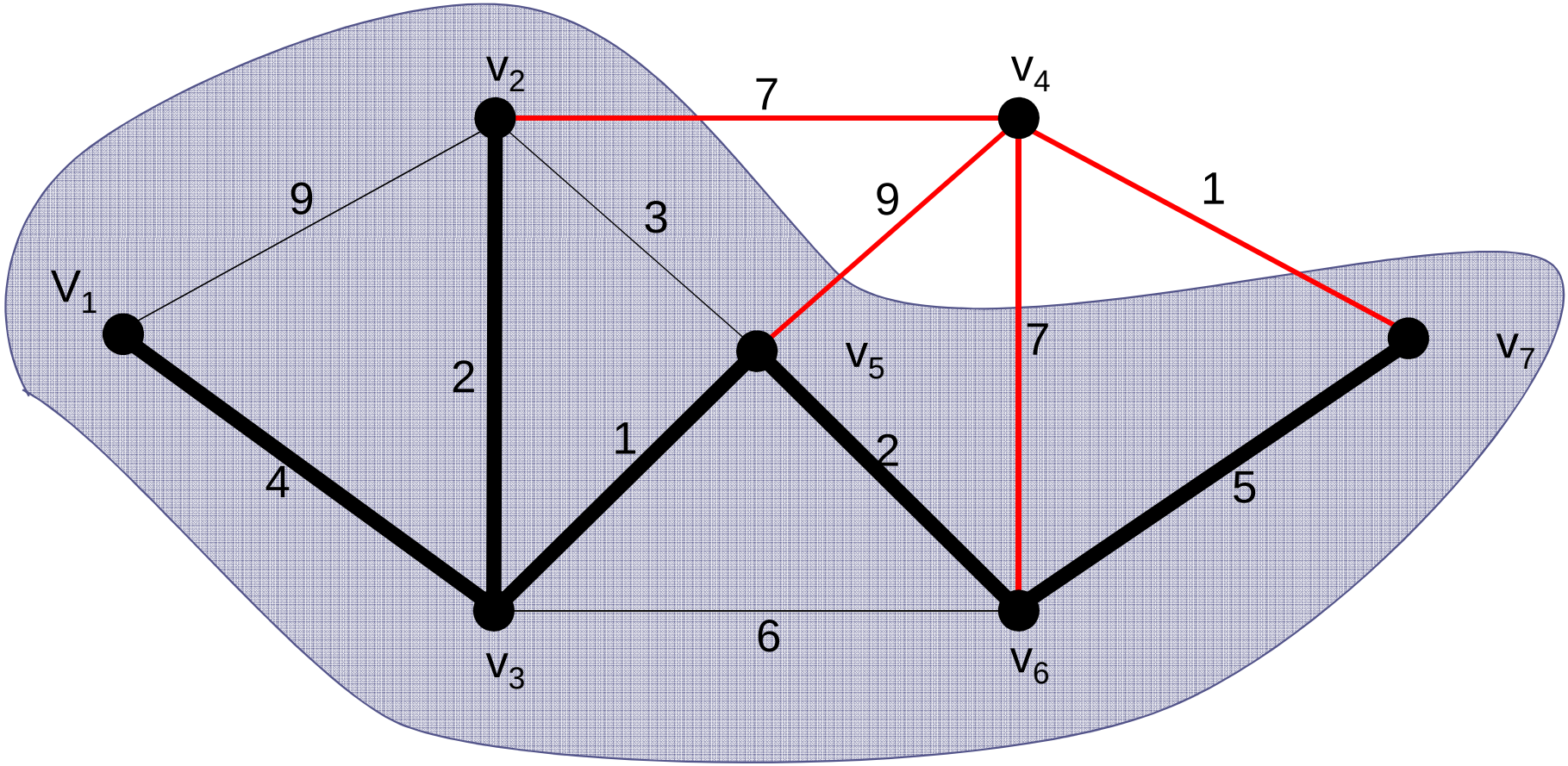
V_2 μπαίνει στο δένδρο

Β. Θεωρία

1. Άπληστοι Αλγόριθμοι

2. Ελάχιστο Συνδετικό Δένδρο (1.Prim-Παράδειγμα Εκτέλεσης)

- Βήμα 5:



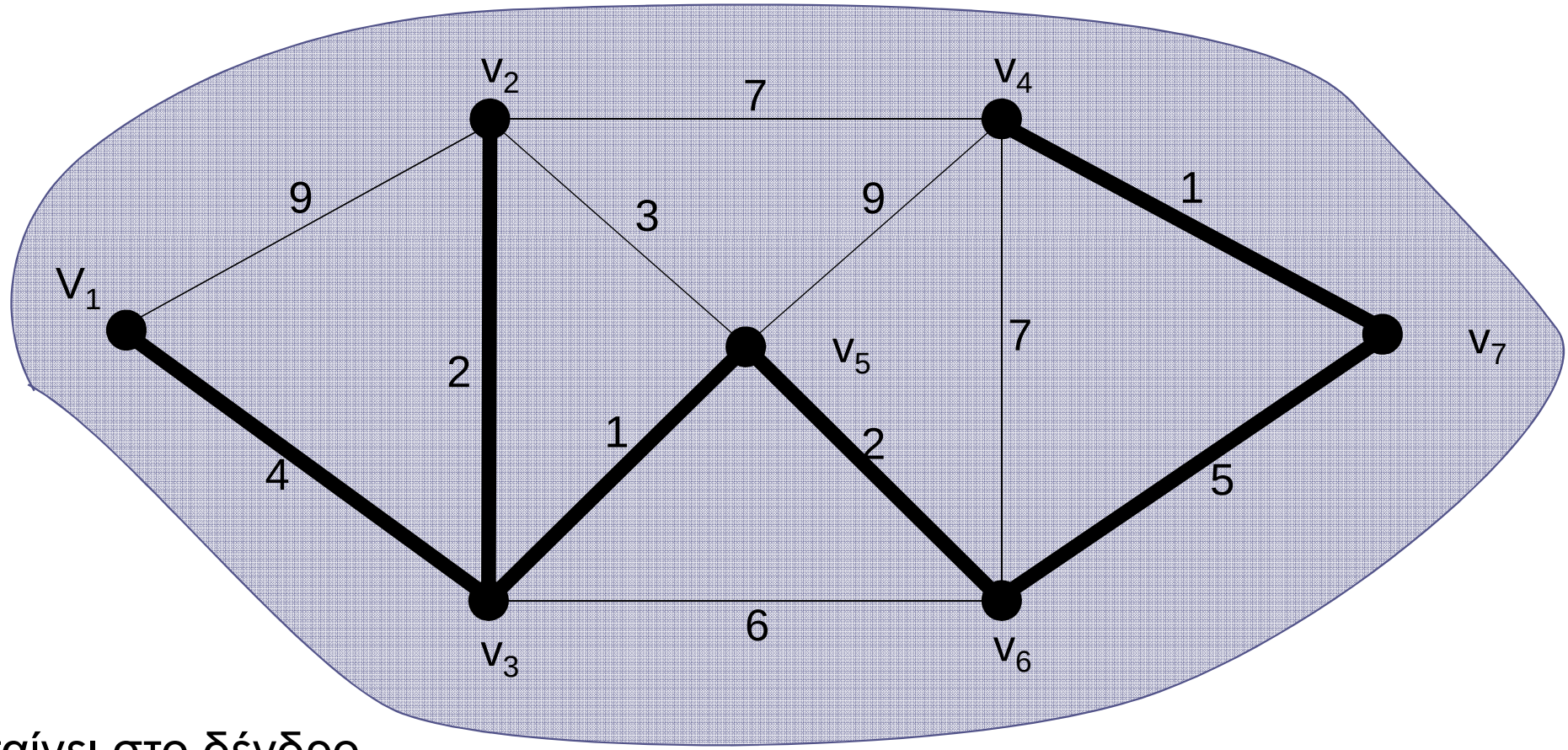
V_7 μπαίνει στο δένδρο

Β. Θεωρία

1. Άπληστοι Αλγόριθμοι

2. Ελάχιστο Συνδετικό Δένδρο (1.Prim-Παράδειγμα Εκτέλεσης)

- Βήμα 5:



V_4 μπαίνει στο δένδρο

Τέλος Αλγορίθμου. Βάρος Ελάχιστου συνδετικού Δένδρου: 4+2+1+2+5+1=15

Β. Θεωρία

1. Άπληστοι Αλγόριθμοι

2. Ελάχιστο Συνδετικό Δένδρο (2.Kruskal-Σκιαγράφηση)

Σκιαγράφηση Αλγόριθμου Kruskal:

- Στην αρχικοποίηση:
  - Θεωρούμε ταξινόμηση των βαρών των ακμών σε αύξουσα σειρά.
- Σε κάθε βήμα:
  - Εξετάζεται η επόμενη ακμή με βάση την ταξινόμηση.
  - Αν δεν δημιουργείται κύκλος εισάγεται στο ελάχιστο συνδετικό δένδρο
- Τερματισμός:
  - Όταν όλες οι κορυφές εισαχθούν στο δένδρο.

Β. Θεωρία

1. Άπληστοι Αλγόριθμοι

2. Ελάχιστο Συνδετικό Δένδρο (2.Kruskal - Ψευδοκώδικας)

```
procedure Kruskal(G=(V,E,W))

   V'=∅

   MergeSort(E) ως προς το βάρος των ακμών

   while |V'|<|V| do
      if E'∪e_i δεν δημιουργεί κύκλο then
          Θέσε E'=E'∪e_i, V'=V'∪{κορυφές της e_i}
      end if
      i=i+1
   end while

   return T=(V',E')

end procedure
```

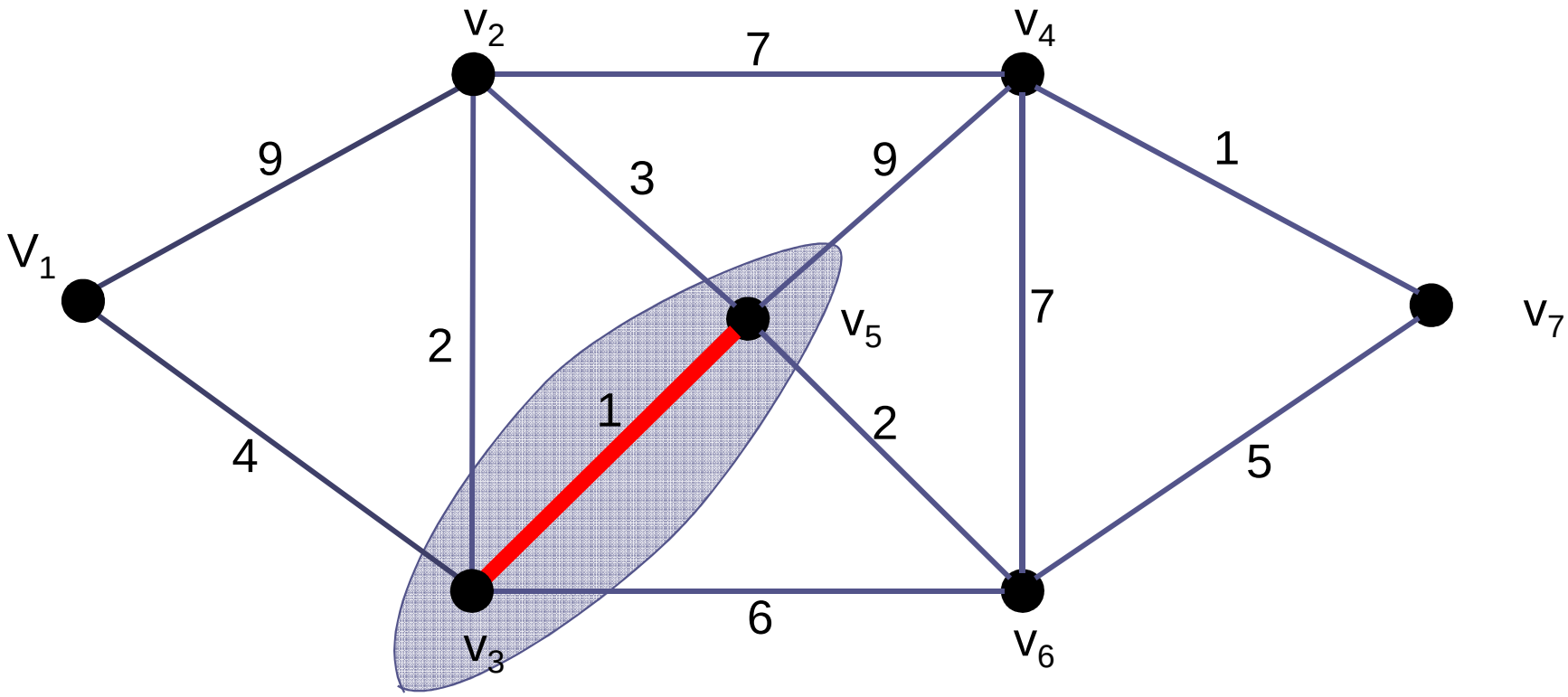
Αποδεικνύεται ότι η πολυπλοκότητα του Kruskal με χρήση κατάλληλων δομών δεδομένων είναι O(m logn)

Β. Θεωρία

1. Άπληστοι Αλγόριθμοι

2. Ελάχιστο Συνδετικό Δένδρο (1.Kruskal-Παράδειγμα Εκτέλεσης)

- Βήμα 1:



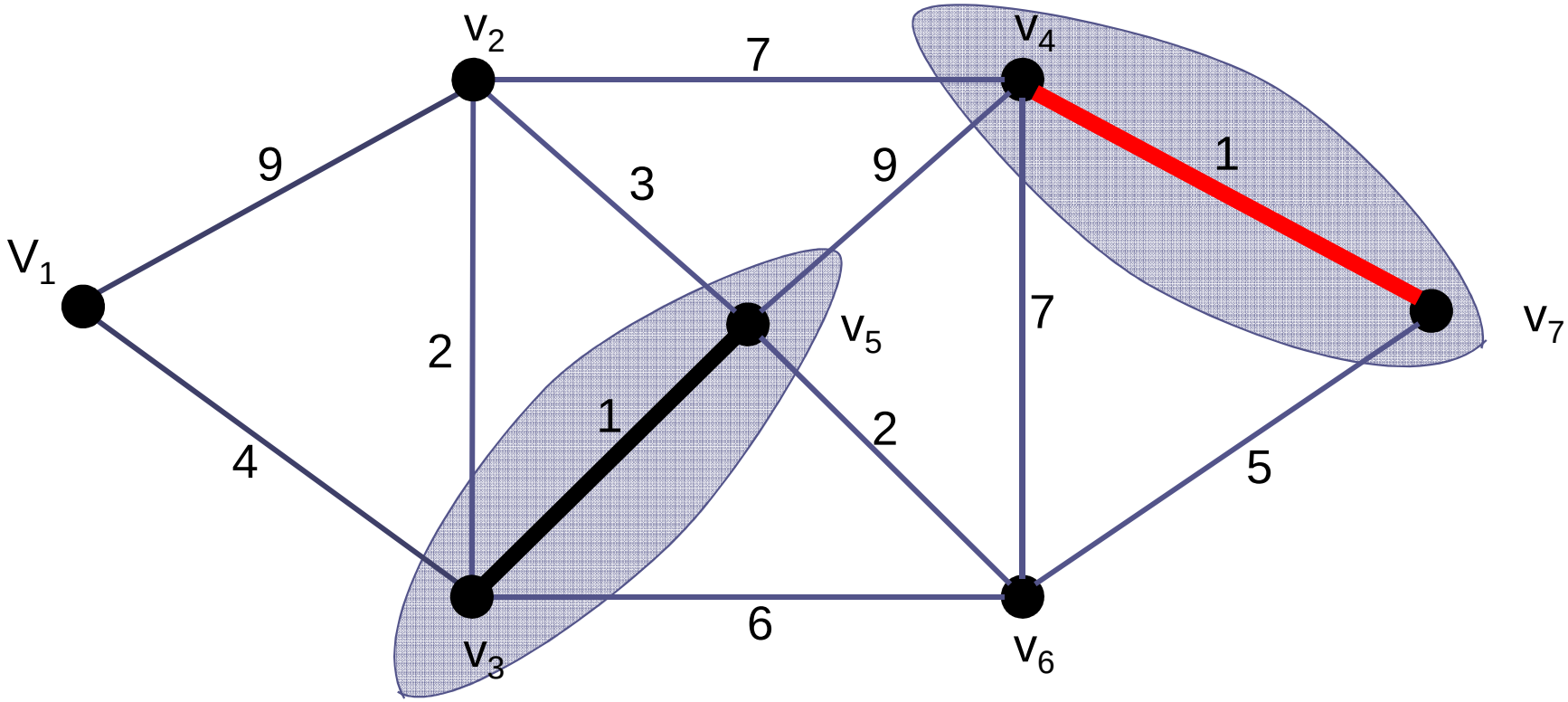
(v_3,v_5) μπαίνει στο δένδρο

Β. Θεωρία

1. Άπληστοι Αλγόριθμοι

2. Ελάχιστο Συνδετικό Δένδρο (1.Kruskal-Παράδειγμα Εκτέλεσης)

- Βήμα 2:



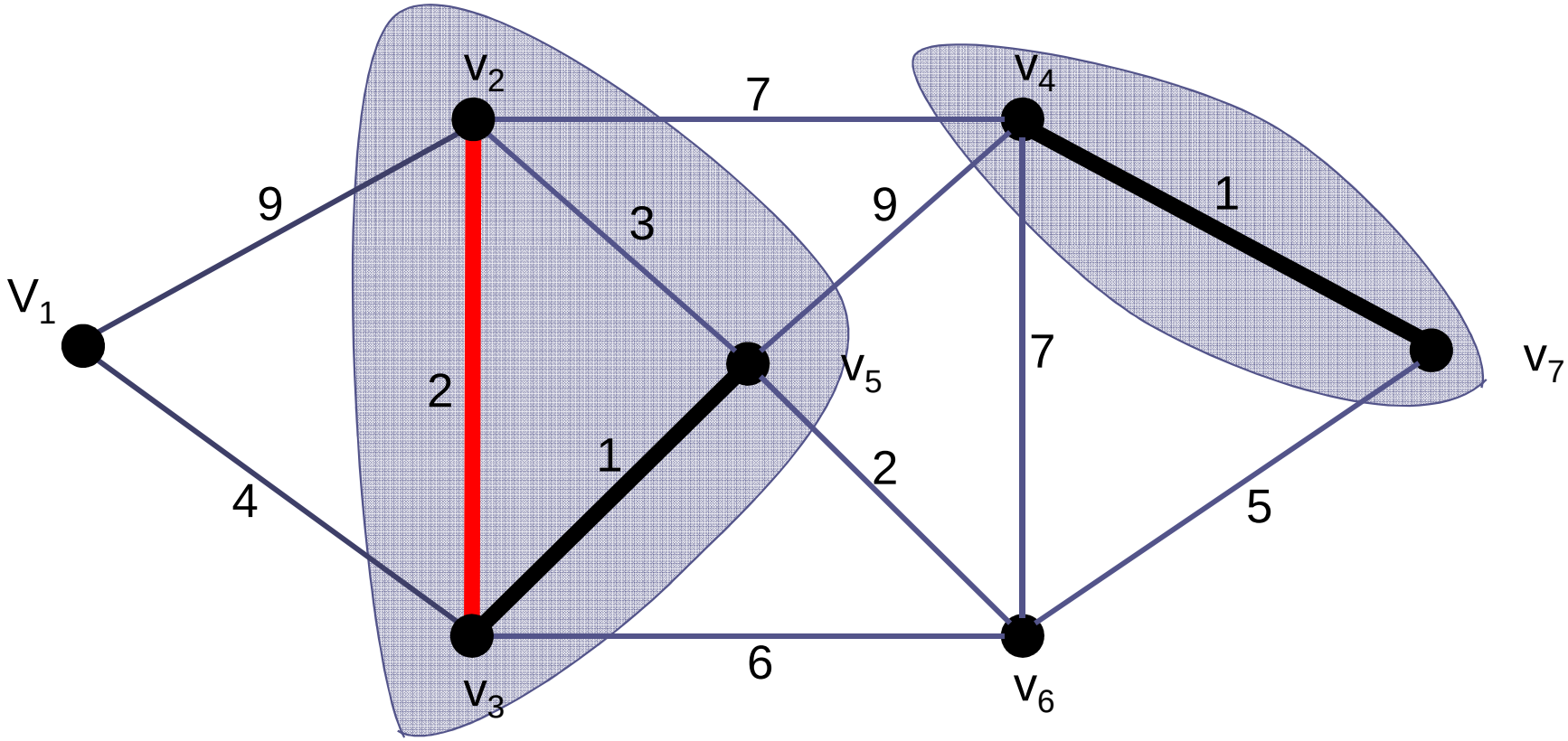
(v_4,v_7) μπαίνει στο δένδρο

Β. Θεωρία

1. Άπληστοι Αλγόριθμοι

2. Ελάχιστο Συνδετικό Δένδρο (1.Kruskal-Παράδειγμα Εκτέλεσης)

- Βήμα 3:



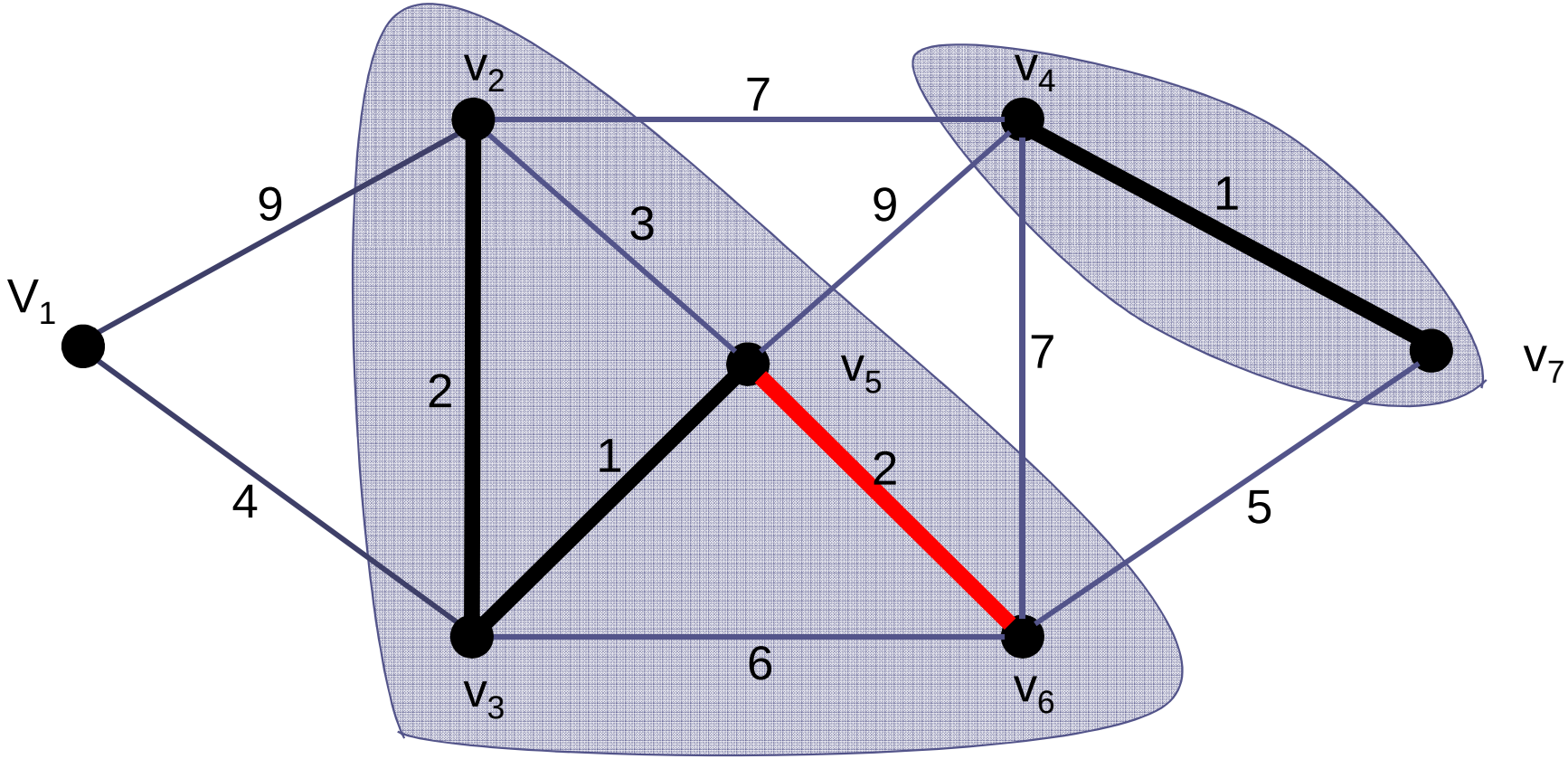
(v_2,v_3) μπαίνει στο δένδρο

Β. Θεωρία

1. Άπληστοι Αλγόριθμοι

2. Ελάχιστο Συνδετικό Δένδρο (1.Kruskal-Παράδειγμα Εκτέλεσης)

- Βήμα 4:



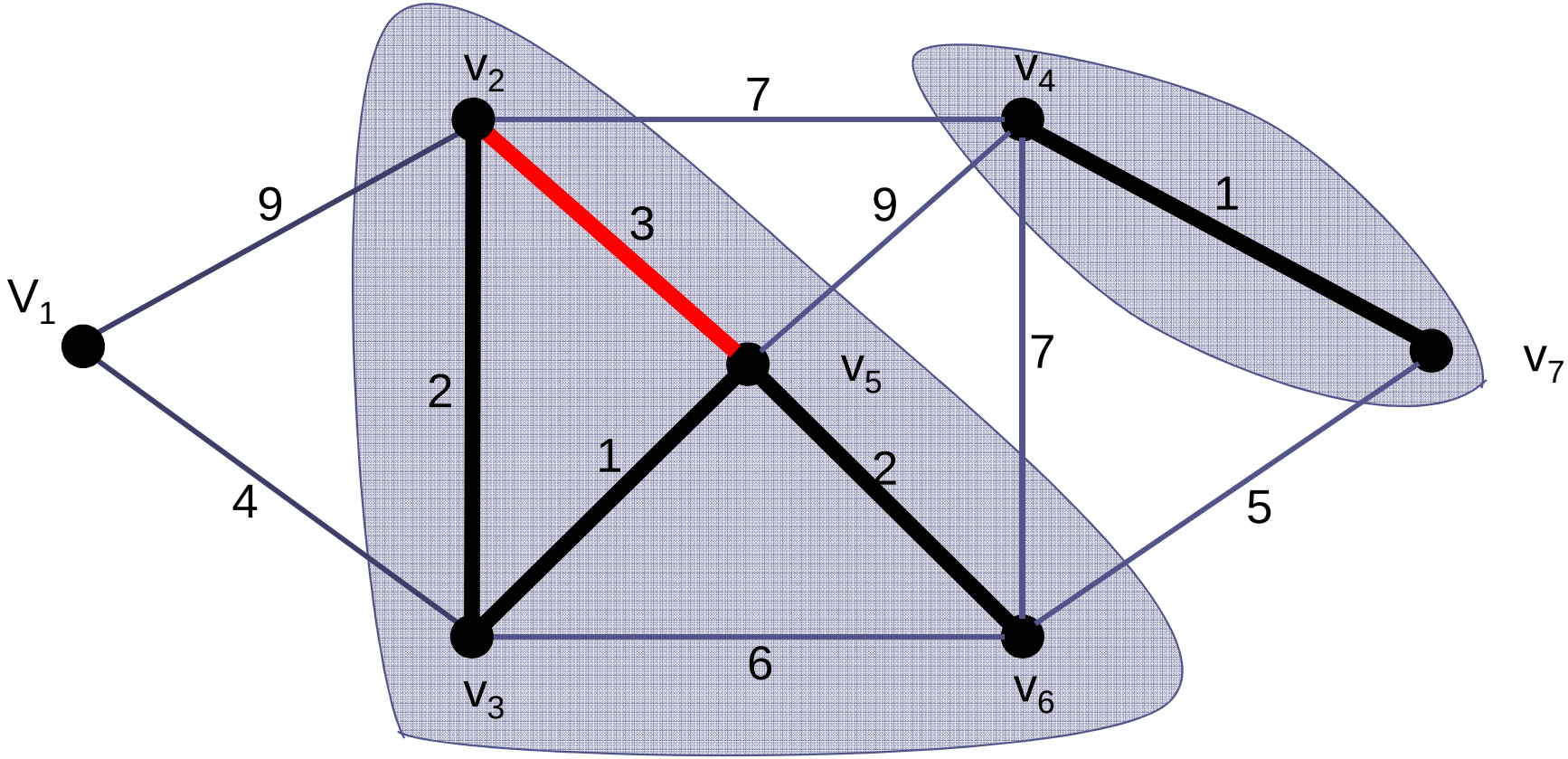
(v_5,v_6) μπαίνει στο δένδρο

Β. Θεωρία

1. Άπληστοι Αλγόριθμοι

2. Ελάχιστο Συνδετικό Δένδρο (1.Kruskal-Παράδειγμα Εκτέλεσης)

- Βήμα 5:



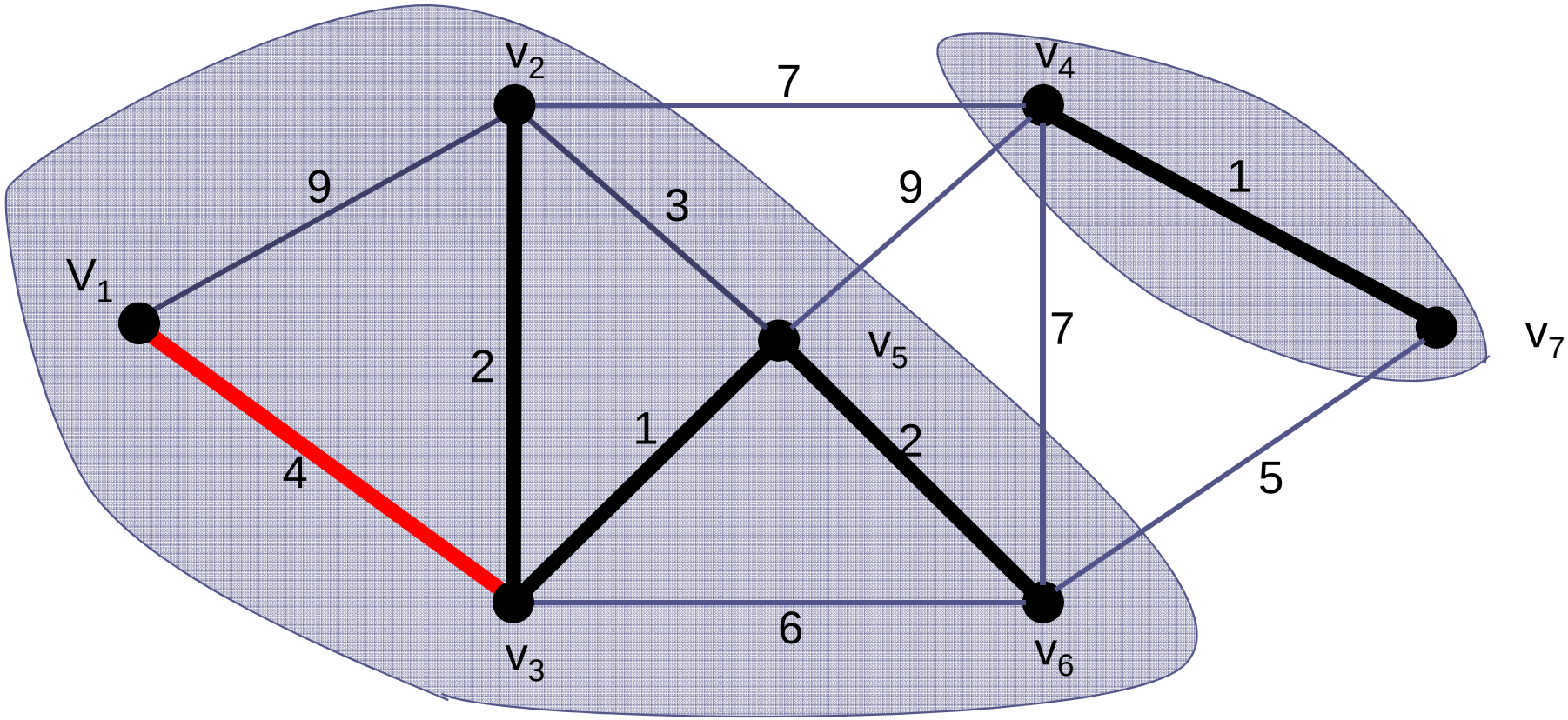
(v_2,v_5) δεν μπαίνει στο δένδρο γιατί δημιουργεί κύκλο

Β. Θεωρία

1. Άπληστοι Αλγόριθμοι

2. Ελάχιστο Συνδετικό Δένδρο (1.Kruskal-Παράδειγμα Εκτέλεσης)

- Βήμα 6:



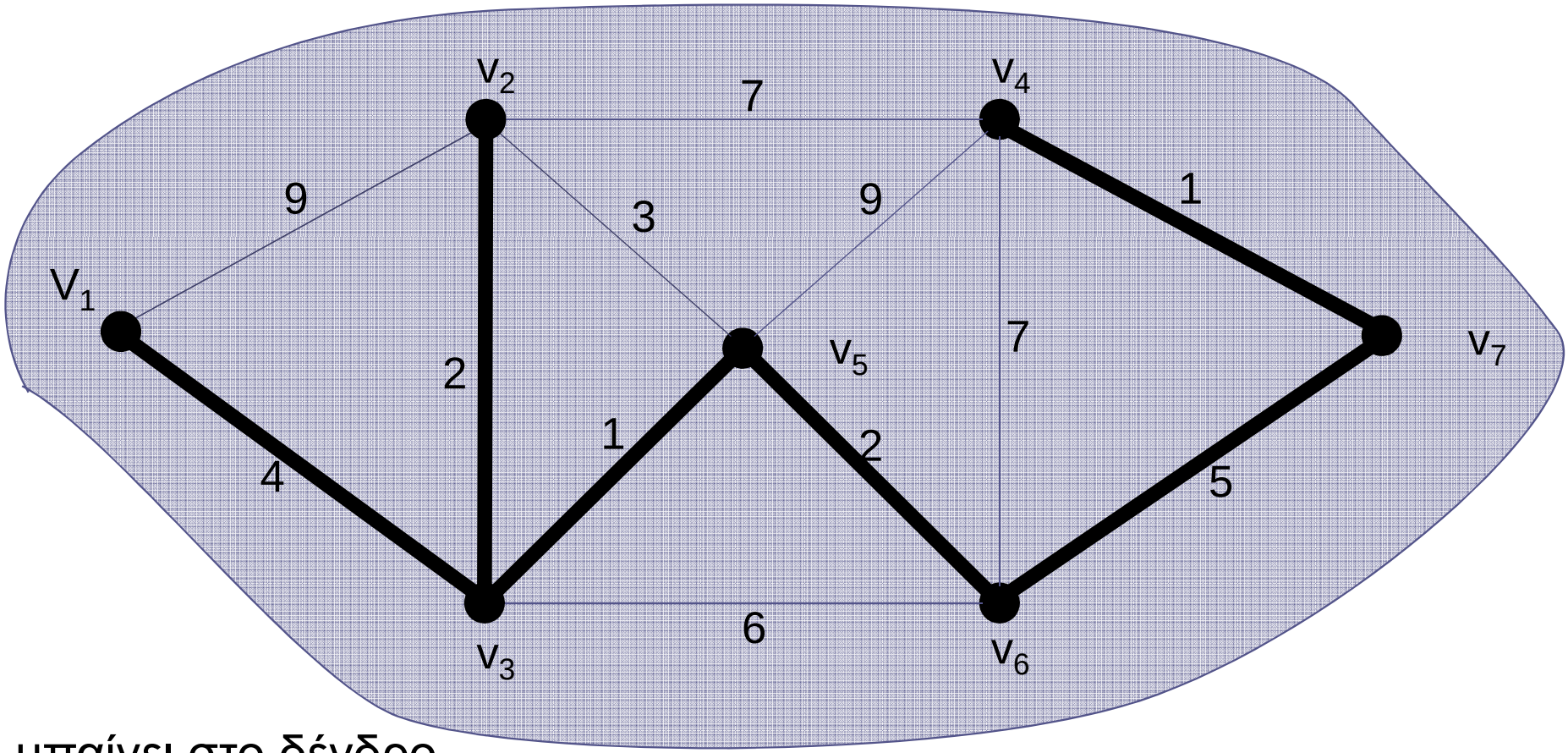
(v_1,v_3) μπαίνει στο δένδρο

Β. Θεωρία

1. Άπληστοι Αλγόριθμοι

2. Ελάχιστο Συνδετικό Δένδρο (1.Kruskal-Παράδειγμα Εκτέλεσης)

- Βήμα 7:



(v_6,v_7) μπαίνει στο δένδρο

Τέλος Αλγορίθμου. Βάρος Ελάχιστου Συνδετικού Δένδρου: 4+2+1+2+5+1=15

Β. Θεωρία

1. Άπληστοι Αλγόριθμοι

3. Επιστροφή Νομισμάτων για Ρέστα

- ΠΡΟΒΛΗΜΑ:
  - Δίδεται ότι έχουμε απεριόριστα νομίσματα 20λεπτών, 5λεπτών και 1 λεπτού. Θέλουμε να επιστρέψουμε ρέστα Χ ευρώ, ελαχιστοποιώντας το πλήθος των νομισμάτων που επιστρέφονται

- Στιγμιότυπα:
  - 1,45 ευρώ: Η βέλτιστη λύση είναι 7 εικοσάλεπτα και 1 πεντάλεπτο.
  - 0,77 ευρώ: Η βέλτιστη λύση είναι 3 εικοσάλεπτα, 3 πεντάλεπτα και 2 μονόλεπτα.

- Άπληστος Αλγόριθμος:
  - Επέλεξε πρώτα όσα περισσότερα 20λεπτα, έπειτα όσο περισσότερα 5λεπτα και έπειτα συμπλήρωνεις με 1λεπτα.

Β. Θεωρία

1. Άπληστοι Αλγόριθμοι

3. Επιστροφή Νομισμάτων για Ρέστα (Ψευδοκώδικας)

```
procedure coins(X, V) //X το ποσό για ρέστα, V: οι αξίες των νομισμάτων

   C=[]
   sum=0
   i=1
   while sum<X do
      Επέλεξε v∈V με μέγιστη αξία έτσι ώστε: sum+v<=S
      C[i]=v
      sum=sum+v
      i=i+1
   end while

   return C

end procedure
```

Πολυπλοκότητα O(X)

Β. Θεωρία

1. Άπληστοι Αλγόριθμοι

3. Επιστροφή Νομισμάτων για Ρέστα (Απόδειξη ορθότητας)

- Έστω ότι θέλουμε να επιστρέψουμε Χ ευρώ και ότι ο άπληστος αλγόριθμος δεν επιστρέφει την βέλτιστη λύση.
  - Εστω OPT=(o_1,o_2,o_3) η βέλτιστη λύση (όπου o_1,o_2,o_3 αντίστοιχα τα πλήθη των νομισμάτων (20λεπτα-5λεπτα-1λεπτα) που επιλέγονται
  - Εστω C=(c_1,c_2,c_3) η λύση του άπληστου αλγορίθμου (όπου c_1,c_2,c_3 αντίστοιχα τα πλήθη των νομισμάτων (20λεπτα-5λεπτα-1λεπτα) που επιλέγονται
- Παρατηρούμε ότι
  - Δεν μπορεί να ισχύει o1>c1 διότι θα σήμαινε ότι έχουν περισσέψει 20 λεπτά και ο αλγόριθμος έχει επιλέξει περισσότερα 20λεπτα. Άρα $o_1=c_1$
  - Δεν μπορεί να ισχύει o2>c2 διότι θα σήμαινε ότι έχουν περισσέψει 5 λεπτά και ο αλγόριθμος έχει επιλέξει περισσότερα 5λεπτα. Άρα $o_2=c_2$
  - Δεν μπορεί να έχουμε c3>o3, διότι με βάση τις προηγούμενες δύο ισότητες απομένει το ίδιο ποσό για να συμπληρωθεί. Άρα $o_3=c_3$
- Επομένως η λύση που επιστρέφει ο αλγόριθμος είναι βέλτιστη.

Β. Θεωρία

1. Άπληστοι Αλγόριθμοι

4. Επιστροφή Νομισμάτων για Ρέστα (Απόδειξη ορθότητας)

- Ωστόσο ο αλγόριθμος δεν είναι πάντα βέλτιστος!!
- Εξαρτάται από τις αξίες των νομισμάτων που έχουμε στην διάθεσή μας.
- Για παράδειγμα αν έχουμε στην διάθεσή μας νομίσματα αξίας 1, 13 και 29 ευρώ και θέλουμε να επιστρέψουμε ρέστα αξίας 39 ευρώ:
  - Ο αλγόριθμος θα επιλέξει 1 κέρμα των 29 ευρώ και 10 κέρματα του 1 ευρώ (σύνολο 11 κέρματα)
  - Η βέλτιστη λύση είναι 3 κέρματα των 13 ευρώ.

Γ. Ασκήσεις

Εφαρμογή 1

Φανταστείτε πως στο ταχυδρομείο υπάρχουν διαθέσιμα γραμματόσημα με τις ακόλουθες αξίες: **1 λεπτό, 5 λεπτά, 9 λεπτά, 10 λεπτά, 25 λεπτά και 48 λεπτά** και δοθέντος ενός χρηματικού ποσού S, ζητάμε το μικρότερο δυνατό πλήθος γραμματοσήμων αξίας S.

1. Σχεδιάστε έναν άπληστο αλγόριθμο για το παραπάνω πρόβλημα.

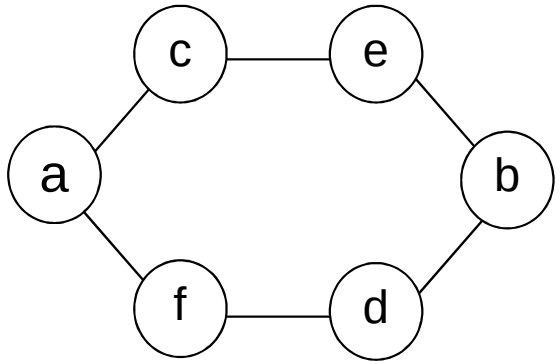
2. Υπολογίστε το χρόνο εκτέλεσης του άπληστου αλγόριθμου.

3. Δώστε τυπική απόδειξη για την ορθότητα του άπληστου αλγόριθμου ή αντιπαράδειγμα για την μη ορθότητα του.

Γ. Ασκήσεις

Εφαρμογή 2

Ένας *έγκυρος χρωματισμός* ενός γραφήματος G = (V, E) είναι μια ανάθεση θετικών ακεραίων (χρώματα) στις κορυφές του γραφήματος G τέτοια ώστε για κάθε ακμή (v_i,v_j) του γραφήματος G να ισχύει χρώμα(v_i) ≠ χρώμα(v_j). Ένας *βέλτιστος χρωματισμός* ενός γραφήματος είναι ένας έγκυρος χρωματισμός που χρησιμοποιεί το μικρότερο δυνατό αριθμό χρωμάτων. Υποθέτουμε ότι οι κορυφές του γραφήματος μας δίνονται με μια συγκεκριμένη διάταξη π = [v_1, …, v_n], όπου n = | V |. Με βάση την διάταξη αυτή έχουμε τον ακόλουθο αλγόριθμο χρωματισμού: για i =1 έως n, ανάθεσε στην κορυφή v_i το μικρότερο δυνατό χρώμα που δεν χρησιμοποιείται από τους γείτονες της v_i μεταξύ των v_1, …, v_{i-1}. Ένα παράδειγμα εκτέλεσης του αλγόριθμου χρωματισμού φαίνεται στο ακόλουθο γράφημα:



π = [a, b, c, d, e, f]

κορυφή:	a	b	c	d	e	f
χρώμα:	1	1	2	2	3	3

Γ. Ασκήσεις

Εφαρμογή 2

(Α) Είναι ο αλγόριθμος χρωματισμού άπληστος; Δικαιολογήστε την απάντησή σας.
(Β) Υπολογίστε την πολυπλοκότητα χρόνου του αλγορίθμου.
(Γ) Αποδείξτε ότι ο αλγόριθμος όταν εφαρμόζεται σε οποιοδήποτε γράφημα παράγει έναν *έγκυρο* χρωματισμό.
(Δ) Αποδείξτε ότι ο αλγόριθμος όταν εφαρμόζεται σε οποιοδήποτε γράφημα *δεν είναι βέλτιστος* (με αντιπαράδειγμα).